COMPUTER SYSTEMS LABORATORY

STANFORD UNIVERSITY · STANFORD, CA 94305-2192

THROUGHPUT PERFORMANCE OF SPREAD SPECTRUM MULTIPLE ACCESS PACKET RADIO NETWORKS

James Scott Storey

Technical Report: CSL-TR-87-319

March 1987

This report is the author's Ph.D. dissertation which was completed under the advisorship of Professor Fouad A. Tobagi. This work was supported by the Defense Advanced Research Projects Agency under Contracts No. MDA 903-79-C-0201 and No. MDA 903-84-K-0249 and a fellowship from the Hughes Aircraft Company.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 87-319	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THROUGHPUT PERFORMANCE OF SPREAD SPECTRUM MULTIPLE ACCESS PACKET RADIO NETWORKS		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL REPORT 6. PERFORMING ORG. REPORT NUMBER 87-319
7. AUTHOR(s) James Scott Storey		8. CONTRACT OR GRANT NUMBER(s) MDA 903-79-C-0201 MDA 903-84-K-0249
9. PERFORMING ORGANIZATION NAME AND ADDRESS Stanford Electronics Laboratory Stanford University Stanford, California 94305-2192		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency Information Processing Techniques Office 1400 Wilson Blvd., Arlington, VA 22209		12. REPORT DATE March 1987 13. NUMBER OF PAGES 217
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Resident Representative Office of Naval Research Durand 165 Stanford University, Stanford, CA 94305-2192		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this dissertation, an integrated model of an unslotted spread spectrum multiple access (SSMA) packet radio network is developed. The model combines a detailed model of the radio channel, accounting for the effect of multi-user interference, with a network model that traces the evolution of the number of interfering transmissons and accounts for the half-duplex nature of the radios. A new analysis of the performance of a Viterbi decoder in a packet radio environment allows the model to be used for a system with convolutional

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

forward error correction (FEC) coding. The synchronization process is also analyzed and incorporated into the integrated model. The model makes use of several approximations for tractability. A discrete event simulation is used to validate these approximations.

The integrated model leads to numerical results which show the throughput and the probability of success for a packet as a function of the channel level and network level parameters. Specifically, for the radio channel, these parameter include the modulation format, the received signal power, the spread spectrum bandwidth expansion, and the presence or absence of FEC coding. Also, the throughput is found as a function of the synchronization parameters, such as the detection thresholds and correlation times, for several different synchronization circuits and preamble structures. At the network level, the parameters are the traffic rate and the network size. The random access protocols considered include unslotted ALOHA and channel load sensing, which is an extension of carrier sensing to an SSMA network.

The results indicate the effects on the throughput of receiver availability, channel errors, and the synchronization process. For each of these effects, regions are found where the effect is the primary factor limiting performance. REsults for the network with a channel load sensing protocol showed little improvement over the results for the network with the simpler ALOHA protocol. Results for the FEC coded and uncoded systems are compared, and it is seen that the systems with FEC coding outperform the uncoded systems. A further finding is that by using a two stage acquisition circuit, the performance can be very close to the performance achieved with an ideal synchronization process.

THROUGHPUT PERFORMANCE OF SPREAD SPECTRUM MULTIPLE ACCESS PACKET RADIO NETWORKS

James Scott Storey

Technical Report: CSL-TR-87-319

March 1987

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305-2191

Abstract

In this dissertation, an integrated model of an unslotted spread spectrum multiple access (SSMA) packet radio network is developed. The model combines a detailed model of the radio channel, accounting for the effect of multi-user interference, with a network model that traces the evolution of the number of interfering transmissons and accounts for the half-duplex nature of the radios. A new analysis of the performance of a Viterbi decoder in a packet radio environment allows the model to be used for a system with convolutional forward error correction (FEC) coding. The synchronization process is also analyzed and incorporated into the integrated model. The model makes use of several approximations for tractability. A discrete event simulation is used to validate these approximations.

The integrated model leads to numerical results which show the throughput and the probability of success for a packet as a function of the channel level and network level parameters. Specifically, for the radio channel, these parameter include the modulation format, the received signal power, the spread spectrum bandwidth expansion, and the presence or absence of FEC coding. Also, the throughput is found as a function of the synchronization parameters, such as the detection thresholds and correlation times, for several different synchronization circuits and preamble structures. At the network level, the parameters are the traffic rate and the network size. The random access protocols considered include unslotted ALOHA and channel load sensing, which is an extension of carrier sensing to an SSMA network.

The results indicate the effects on the throughput of receiver availability, channel errors, and the synchronization process. For each of these effects, regions are found where the effect is the primary factor limiting performance. REsults for the network with a channel load sensing protocol showed little improvement over the results for the network with the simpler ALOHA protocol. Results for the FEC coded and uncoded systems are compared, and it is seen that the systems with FEC coding outperform the uncoded systems. A further finding is that by using a two stage acquisition circuit, the performance can be very close to the performance achieved with an ideal synchronization process.

Acknowledgments

There were many people who have helped me during my years at Stanford, both professionally and personally. First, I would like to thank my advisor, Fouad Tobagi, for introducing me to the topic of packet radios, and for his assistance and advice over the past several years. Also, I appreciate the efforts of the other committee members, Thomas Cover, Richard Swanson and John Cioffi. No work proceeds in complete isolation, and my research benefited greatly from the suggestions and criticisms of the other members of the FAT group. I am also thankful to Roger Strauch, whose advice and friendship helped guide me into and through the Ph.D. program. A number of other friends also deserve recognition for making the years at Stanford more pleasant. In particular, I would like to thank Bob Wentworth, Philip Kegelmeyer, Dave Meyer, and Peter Arnold. I received a great deal of encouragement and support from my mother and my sister, and I know that they are as happy for me as I am for myself. Finally, my greatest thanks go to Donna George, who put up with my most difficult year and made it into the best year of all.

Contents

Tables

Figures

Chapter 1

Introduction

1.1 Introduction

Over the past several decades, digital information has constituted an ever increasing proportion of the traffic over telecommunications systems. There are two main factors contributing to this trend. First, along with the explosive growth in the number of computers, there has been an increase in the amount of data being sent to, from, and between computers. Second, with the emergence of low cost implementations made possible by VLSI, it has become economical to digitally encode a variety of analog signals such as voice, facsimile, and video, usually resulting in lower cost and higher quality systems. The need for digital communications to support both computer data and digitized analog signals has spawned a great deal of research into efficient means for the transmission of digital data.

The characteristics of computer communications differ from those of voice traffic. One difference is that in computer networks there is often a requirement for a

very low probability of transmission error. For example, noise in a voice conversation can lead to static and a perceived degradation of quality, while similar noise occurring during the transmission of financial records could cause a loss of millions of dollars. Another characteristic of computer networks is that the traffic from individual users comes in bursts. More precisely, there are long periods during which no data is generated (idle time), punctuated by short periods during which large amounts of data are generated for transmission. This characteristic of the traffic is referred to as burstiness. As an example of why the traffic is bursty, consider the communications between a remote line-oriented terminal and a central computer. The link from the terminal to the computer is idle while the person at the terminal ponders what to do next. When he finally types in a line of data followed by a carriage return, the entire line of up to 80 characters is transmitted, after which the link from the terminal to the computer is idle until the next line. Because the characteristics of computer traffic are different from those of the more traditional voice traffic, new techniques were developed to allow the communications resources in a computer network to be used efficiently.

One such technique is the use of error detection to insure that transmission errors are identified. Network protocols include a means for notifying the source of a message when an error in the received data is spotted so that the message can be repeated. However, this method of repeating unsuccessfully transmitted messages can be very wasteful, as a single error can cause a long message to be repeated. The efficiency can be improved by dividing the messages into shorter blocks of data called packets, typically about 1000 to 4000 bits in length. Unless the channel is very bad, most of the packets will be received error free, so only the fraction of the message corresponding to the unsuccessful packets need be repeated.

The actual transmission medium for the packets can take a variety of forms.

The most common forms include copper wire, coaxial cable, fiber optics, microwave point to point links, satellite channels, and broadcast radio. In this dissertation, the performance of a packet network using a broadcast radio channel, i.e., a packet radio (PR) network, is analyzed. There are several reasons why radio might be chosen as the transmission medium. If there are mobile users, or if the network must be deployed quickly, then radio is the only feasible choice. Many of the military systems have these criteria, so much of the work on PR networks has been sponsored by the military. The choice of radio might also be made if the average aggregate data rate that must be supported is small. For such a network, transmission via radio may be the lowest cost alternative. An example of such a system is the Stanford Packet Radio Project [18].

The use of radio requires a scheme for providing multiple access capability, that is, the ability to share the limited bandwidth among the users of the network. The simplest method is to split the bandwidth into channels and assign a channel to each link, i.e., each source-destination pair with a non-zero traffic level. This is referred to as frequency division multiplexing (FDM). If the radios can synchronize to a common clock, the users can take turns using the entire bandwidth, in a time division multiplexing (TDM) system. These static allocation schemes can be very efficient if there is a constant level of traffic over each link. However, as discussed previously, in a computer network, the traffic is not constant but is bursty, so only a small fraction of the frequency channels or time slots will be in use at any time. The radio bandwidth can be utilized more efficiently if the channels or time slots are allocated only to those radios with traffic to send. In a demand assignment multiple access (DAMA) system, a small portion of the radio channel is set aside for reservations. A radio with traffic to transmit first sends a reservation request. If an open channel is available, a channel is allocated to that radio. This is similar to the public telephone network, where a channel is assigned automatically

when the receiver is picked up. With demand assignment, the network traffic can be supported by a much smaller total bandwidth. The only drawbacks are the complexity of the reservation scheme and the possibility that the network will be busy when a reservation request is made, causing a call to be rescheduled for later.

In a computer network, even with demand assignment, if a channel is allocated for the entire length of a call the channel may be idle for long stretches of time. This problem is alleviated by the use of message or packet switching. With message switching, the channel is only allocated for the duration of a message. Thus, in the example of an interactive session presented earlier, a channel is allocated for traffic from the terminal to the computer only for the short intervals following each carriage return during which there is a message to send. Taking this one step further, in a packet switched DAMA system, each packet requires channel allocation. The use of message or packet switching can greatly reduce the percentage of time that allocated channels sit idle.

Although the available bandwidth will be busy a large fraction of the time, the overhead required for channel allocation can be significant. Therefore, many packet switched systems make use of a random access scheme for sharing the channel. In a random access system, there is little or no overhead for channel allocation. The simplest random access scheme is ALOHA [1]. Each radio with a packet to send independently chooses a random point in time at which to transmit. At the transmission time, referred to as the scheduling point, the packet is transmitted using the entire available bandwidth. If two packets overlap in time, a collision is said to occur, and both packets must be retransmitted. As long as the traffic level is low, the probability of a collision will be small, so the waste due to retransmissions will also be small. If the radios are half-duplex, meaning they cannot transmit and receive at the same time, a slightly modified version known as disciplined ALOHA

can be used [27], [5]. With this protocol, a packet is transmitted at the random scheduling point only if the radio is not busy receiving a packet. Thus, a radio will not throw away a partially received packet in order to transmit a packet. Because they are matched to the characteristics of the traffic, packet switched random access networks allow the communications resources to be used efficiently for the transmission of computer data.

In addition to the necessity of providing a multiple access scheme, there are various other drawbacks to using radio, especially for the military user who has the additional constraints of security and survivability. To overcome the two most serious problems, namely, jamming and interception of transmissions, spread spectrum signalling was developed. With this type of signalling, the transmitted signal is spread over a very wide bandwidth so that the energy density per Hertz is very small. This results in a signal that is robust in the presence of hostile jammers, and one that is even difficult for an enemy to detect. Almost all of the early work on spread spectrum was classified, and the technique was used only in military systems. In addition to the anti-jam and anti-intercept properties, spread spectrum signalling provides several other benefits. These include the capability of ranging, robustness against multi-path, and the possibility of spread spectrum multiple access, discussed below. Consequently, over the past two decades, a number of unclassified papers and books on spread spectrum have been published, and commercial systems making use of this technique have emerged. Scholtz has written a detailed account of the history of spread spectrum in [45].

In a multiple access network, spread spectrum channels have the additional benefit of allowing the successful transmission of concurrent packets. For systems in which spread spectrum is not used, commonly called narrowband systems, any packets which overlap will, with a probability very nearly equal to one, contain

errors and therefore will not be successful. In a spread spectrum system, if different spreading codes are used, a number of transmissions can simultaneously occupy the channel, and each will suffer only slight degradation in signal quality compared to the performance of a single transmission. Consequently, when an ALOHA transmission protocol is used in a radio network with sufficient spreading, more than one packet can be transmitted concurrently with a non-zero probability of success. Such a system is referred to as a spread spectrum multiple access (SSMA) system. Furthermore, if the codes being used are roughly orthogonal, the probability of success will be almost one. A system which relies on the low cross-correlation of an orthogonal set of codes is referred to as a code division multiple access (CDMA) system.

The technique of forward error correction (FEC) coding can improve the performance of an SSMA system. In a system using FEC coding, the data is formed into codewords which are sequences of binary symbols, such that there are more symbols in a codeword that there are bits in the data processed. The codewords form a small subset of all possible combinations of symbols. The receiver includes a decoder, which chooses the valid codeword that is closest to the received sequence of symbols. Even if a small percentage of the symbols are received in error, the correct codeword can still be chosen, so there will be no errors in the data bits output by the decoder. Of course, if there are a large number of symbol errors, the wrong codeword will be chosen, and the packet will be unsuccessful.

There are two main classes of FEC coding. In the first class, block coding, the codewords are generated from contiguous blocks of data, typically on the order of 64 or 128 bits. The decoder then processes the entire received block of symbols, finding the fixed length codeword which is closest. In the second class, convolutional coding, the codewords are formed as a running function of the data bits over a fixed

size window. For example, in the convolutional code studied in Chapter 5, the two symbols output at each bit position are a function of the data over the previous seven bits. In effect, a single codeword is generated from the entire packet. The decoder must compare all possible codewords of a length equivalent to the length of the packet, and choose the codeword that is closest to the received sequence. Although this may sound like an impossible task, an algorithm introduced by Viterbi performs precisely this function [51]. The Viterbi algorithm has been implemented with low cost hardware, which has furthered the popularity of convolutional coding with Viterbi decoding.

1.2 Existing Work

In a CDMA system with a well chosen set of spreading codes, the effect of multi-user interference can be reduced relative to the levels encountered in an SSMA system. The evaluation of the bit error rate (BER) of CDMA systems has been a very active field of research. Results for direct sequence (DS) spread spectrum channels which specifically account for the cross-correlations of the codes being used include bounds on the BER (e.g. [38], [39], [14], [29]), and various approximations to the BER (e.g. [53], [19], [20]). These results can also be used to find the BER for SSMA systems by assuming that the codes used are independent random sequences. The expression for random codes are usually much simpler than the expressions accounting for the cross-correlation of codes.

If the multi-user interference is constant during the transmission of a packet, for example in a slotted PR network, the BER results can be used directly to find the probability of a packet being successful (i.e., received without errors). Furthermore, the performance of slotted systems which include (FEC) coding can be found by

using standard techniques for the evaluation of decoder performance. One of the first such analyses was that of Raychaudhuri [43], who derived the throughput and delay performance of a slotted packet network. A frequency-hopped spread spectrum slotted PR network was analyzed by Pursley in [40].

It is feasible to implement a slotted network only if for every radio in the network there is only a small difference in propagation delay between all transmissions reaching that radio. However, for terrestrial PR networks which cover a large geographical area, the difference in propagation delay makes time-slotting difficult. Also, if the radios are mobile, time-slotting can be impossible. Consequently, a large class of PR networks will be unslotted. In order to evaluate the performance of an unslotted network, it is necessary to consider the reception of packets for which the multi-user interference varies over time. This variation causes several difficulties in the analysis. First, the model must account for the number of simultaneous transmissions occurring at each time during the packet reception. Second, for systems using FEC coding, the probability of error depends upon the multi-user interference over an interval of time. Standard results for FEC decoder performance assume a constant error rate. Consequently, the decoder performance must be reevaluated for the case when the error rate varies during the reception of a packet.

An early analysis of an unslotted network, in which a simple model was used to account for the varying multi-user interference, was carried out by Musser and Daigle [31]. From the results of analyses of BER, such as those mentioned earlier, it was seen that for systems using FEC coding, when the number of simultaneous transmissions is constant during the entire packet, the probability of correctly decoding a packet is a very sharp function of the number of transmissions. Musser and Daigle approximated this function by a step function, and considered that a given packet will be successful if the maximum number of simultaneous transmissions oc-

curring during the transmission time of the packet is not greater than some cutoff L, and will be unsuccessful if the maximum is greater than L. By assuming that this step function approximation is valid even when the number of simultaneous transmissions varies, they found the throughput of an unslotted network with an infinite population.

Pursley derived a more exact analysis of a frequency hopped spread spectrum unslotted network with FEC coding [41]. By deriving expressions for the probability distributions of the maximum and minimum number of simultaneous transmissions occurring during a given time interval, he was able to find upper and lower bounds on the probability of success for a fixed length packet which led to bounds on the throughput. Taipale analyzed the performance of a Viterbi decoder and derived a bound on the probability of a packet being successful for a direct-sequence spread spectrum system using convolutional FEC coding that was shown to be tighter than previous bounds [50]. However, this bound requires a constant error rate throughout the entire packet. Using the bounding approach developed by Pursley, he also found bounds on network throughput.

In addition to the occurrence of bit errors due to thermal noise and multi-user interference, there are a variety of other factors which influence the success or failure of a packet transmission. One very important factor is the preamble synchronization process. There has been a great deal of work recently concerning the synchronization of spread spectrum codes. Polydorous and Weber formulated a unified approach to the analysis of synchronization in [35] and [36]. This was extended to the case of a system with modulation and jamming by Siess and Weber in [48]. Although these analyses are aimed at acquiring a steady stream of traffic, rather than a short packet, the results can also be applied to packet radio systems. There have been very few papers published regarding the effect of synchronization in packet systems.

Davis and Gronemeyer used a simple capture model of the synchronization process in a slotted system [11]. They assumed that there is a vulnerable period at the start of the packet such that synchronization will be successful with probability one if no other packets begin transmission within the vulnerable period, and with probability zero if one or more other packets begin in this period. In [44], Rappaport and Schilling consider the acquisition of packets in a frequency hopped system in the presence of jamming. Kowatsch [28] describes a hardware implementation of a synchronization circuit, and presents both theoretical and experimental performance. Despite the number of papers concerning the synchronization of spread spectrum signals, few of these have considered the performance of packet systems, and there have been no published results which precisely account for the effect of multi-user interference on the synchronization process.

In SSMA networks with a large spreading factor or a low density of radios, the multi-user interference is almost negligible. Therefore, such networks can be modeled by assuming that the interfering transmissions of other radios have no effect on the quality of the channel between any pair of radios. If the received signal power is high enough that thermal noise can also be ignored, which is usually the case, then it can be assumed that any packet that is received is successful with probability one, irrespective of the number of radios transmitting. This is referred to as the perfect capture assumption. Recently, there have been a number of studies of the performance of PR networks with perfect capture ([32], [5], [4], [12]). However, for a given data rate, larger spreading factors require a proportionally larger bandwidth. In other words, for a fixed bandwidth and data rate, the multi-user interference will become significant as the number of simultaneous transmissions increases. Since a greater overall throughput may be possible with more transmissions, the maximum performance might be achieved in the range for which perfect capture is not a valid assumption.

In summary, many advances have been made into the analysis of the various parts constituting an SSMA PR network. However, to a large extent, each work has focused on characterizing some effect in isolation from the rest of the system. On the one hand, it may not be clear from the derived results such as BER or probability of synchronization what the impact on network throughput will be. On the other hand, the throughput analyses often ignore or greatly simplify the effects of the channel performance. Another shortcoming is that there is no indication of the interaction between the various effects, or the regions where one effect dominates the overall performance. Clearly, there is a need for a more comprehensive model of a spread spectrum multiple access packet radio network.

1.3 Overview

The aim of this dissertation is to study the performance of a spread spectrum multiple access (SSMA) packet radio (PR) network, and to examine the effect on the network level performance of the performance of the receiver and the SSMA radio channel. To achieve this, it is necessary to combine a number of different aspects of the system into a single model. An exact analysis at every level would lead to a model which is either intractable or is too complex to yield numerical results. Consequently, the analysis requires several approximations so that numerical results can be found. However, the use of crude approximations would lead to a model which does not accurately demonstrate the effects of the channel level parameters. In some instances, it is necessary to rely on bounds rather than tight approximations. Thus, the results are close approximations to lower bounds. Of course, this is neither a true bound nor a tight approximation. Nevertheless, the approximations are much tighter than the bounds, so the result can be considered to be a lower bound, albeit one that results from an approximate analysis.

The formulation of the system model requires a balance between accuracy and tractability. The result is an integrated model, which accounts for the important details of both the radio channel performance and the network evolution over time. At the network level, the half-duplex nature of the radios is specifically included. Furthermore, the model traces the evolution over time of the number of interfering transmissions, and the results account for the probability of success of a packet given this varying level of interference. Specific network level parameters are the network size M and the traffic rate g.

In order to extend the model to study a system using FEC coding, a new analysis of the performance of the Viterbi decoder is required. Standard results treat a system in which the probability of error on the raw channel is constant over all time. In addition, the standard analyses usually lead to the long term average rate of bit errors, rather than the probability of success for a short packet. The analysis in this dissertation extends the existing studies of the performance of the Viterbi decoder to the more general case of a system with a varying channel error rate, and leads to a bound on the probability of success for a packet. The resulting bound allows the integrated system model to be used for a network with FEC coding.

Existing studies of the performance of the spread spectrum code synchronization process are extended to the case of a multiple access system. An approximate analysis leads to the probability of false alarm and probability of detection for a signal received in the presence of interfering transmissions. This is perhaps the first detailed treatment of the synchronization process in a multiple access system. In addition to these probabilities, the overall performance of a specific synchronization circuit is studied, and the model of the synchronization process is incorporated into the integrated system model.

This approach leads to numerical results for the throughput as a function of the the modulation format, the received signal power, E_b/N_0, and the bandwidth expansion W (in chips per bit), for a system using FEC coding as well as for an uncoded system. The derived throughput is also a function of the synchronization circuit used, and of parameters of the synchronization circuit, such as the detection threshold and correlation time. Numerical results indicate three major effects: channel errors, receiver availability and imperfect synchronization. It is shown that with certain synchronization circuits, when the thresholds are optimized, the performance can be very close to the performance of an ideal network with perfect synchronization. However, the performance is fairly sensitive to the parameter settings. Nevertheless, results for the ideal network will give a good approximation to the performance of the realistic system in which the synchronization parameters are optimized. For such an ideal network, it is seen that when the bandwidth expansion W is low, the limitation on throughput is due almost entirely to the channel errors. For such a case, a simpler infinite population model is very accurate. When W is large, the limitation is due to the lack of idle receivers, and the performance is very close to the performance achieved when the channel is perfect. It is also shown that the transition between the two regimes is abrupt, especially for the FEC coded channel. The numerical results show the impact of the three major effects, and the regions where each effect is the primary factor limiting the performance.

The overall system model is described in Chapter 2. This includes a general description of the system, and the assumptions and approximations used to lead to a tractable model. The uncoded radio channel model is summarized in Chapter 3. There are a number of formats that may be used for modulating the RF carrier. In the literature, the probability of error has been determined for many of these formats. Because the results of later chapters rely in part on these radio channel results, a model of the SSMA radio channel and the resulting probabilities of bit

error are reviewed. Modulation formats considered include binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), differential phase shift keying (DPSK), and binary frequency shift keying (FSK). In addition, an upper bound on the probability of error for a BPSK system is described. Pursley [42] has shown that this bound leads to channel errors which are conditionally independent, given the number of simultaneous transmissions. This conditional independence is necessary in the derivation of the FEC decoder model in Chapter 5.

The network model is analyzed in Chapter 4, leading to expressions for the throughput and the average probability of success. Both an infinite population and a finite population model are developed. For the latter model, the number of radios receiving packets is specifically accounted for, giving a more accurate representation of a half-duplex radio network. The models take as parameters the probability of receiving (synchronizing to) a packet and the probability of bit errors in the presence of multi-user interference and noise. Numerical results are given for the uncoded channel, with the assumption of perfect synchronization. The throughput is plotted as a function of the channel parameters, such as received signal power E_b/N_0 and spreading ratio W, and the network parameters, such as traffic rate g and network size M.

In a narrowband system, the network performance can often be improved by using channel sense multiple access (CSMA) [26]. For this CSMA protocol, radios sense the state of the channel and block transmissions when the channel is busy. Similarly, in an SSMA network where multiple users can transmit simultaneously, improvement may be possible if the radios can sense the number of radios transmitting and block transmission when the channel is heavily loaded. This is referred to as channel load sensing. A modification of the network model accounts for a channel load sensing protocol. Numerical results given in §4.3 demonstrate the improvement

due to this protocol.

The performance of a Viterbi decoder in a packet environment is analyzed in Chapter 5, leading to a model of the coded channel which is easily incorporated into the network model. In a system with FEC coding, a packet will be successful only if there are no errors in the decoded bit sequence. The bit errors in the data stream output from a Viterbi decoder are highly correlated, even if the symbol errors are independent. Because there is positive correlation, a simple lower bound on P_S, the probability that a packet is successful, can be found from the product over all bits of one minus the probability of bit error. However, this bound is fairly loose and in Chapter 5, a tighter bound on P_S is derived. The bound is a product over all bits of a bound on the probability that no decoder error is made at the given bit. The bound yields an error process which is independent from bit to bit. Furthermore, it is shown that for the packet systems considered, the bound on the probability of not making a decoder error at a given bit is, to a good approximation, a function only of the current probability of symbol error. This approximate probability is evaluated numerically for a specific rate 1/2, constraint length 7 convolutional code. The validity of the approximation is verified with a simulation model. The analysis yields $P_C(X)$, the probability of a correct decoder decision given X simultaneous transmissions. This is used in the network model to find the throughput of the system with FEC coding. Numerical results indicate the performance of the coded system as a function of the same parameters studied in the uncoded system. Also, the improvement due to FEC coding is shown.

In Chapters 3 through 5, it is assumed that the synchronization process is perfect, so that any packet which is destined to an idle receiver is acquired by that receiver. In Chapter 6, a model of the synchronization process in a multi-user packet radio network is derived. This model is incorporated into the system model in

order to find the degradation due to imperfect synchronization. From the numerical results, the performance of several synchronization circuits can be compared. Also shown are the effects on network throughput of the synchronization parameters, such as the detection threshold and the correlation time.

The integrated model developed in this dissertation accurately models a PR network at many levels. The performance measures derived are close approximations to lower bounds on the throughput and the probability of a packet being successful. These results are found as a function both of radio channel parameters, such as the received signal power and spreading factor, and also of network parameters, such as the number of radios and traffic rate.

Chapter 2

System Model

This chapter is a description of the elements and organization of the specific system being studied, and the modelling assumptions used. A more thorough introduction to PR networks is given by Kahn et al. in [23]. The book by Simon et al. [49] gives an very thorough treatment of spread spectrum systems. The reader is also referred to the tutorial paper by Scholtz [46].

2.1 System Description

The following terminology is used. Bits refer to the uncoded information stream generated by the user. Symbols refer to the output of the error correction encoder (for a rate 1/2 code, there will be 2 symbols per bit). Chips refer to the product of the data sequence and the spread spectrum code sequence, as in [49].

The network under consideration consists of a number of packet radio units (PRU's) sharing a single radio channel. Each PRU contains a transmitter and a receiver, and is connected to a terminal. A radio can be transmitting, receiving,

or idle, but cannot transmit and receive at the same time. Traffic in the network consists of packets that are generated at one terminal, transmitted by the associated PRU, received by the destination PRU, and finally output to the destination terminal. The packets are of varying length, and the network is unslotted. The case of a fully connected single-hop network is considered. Thus, each receiver can hear every transmitter, and there is no need for relaying or store-and-forwarding of packets.

Information to be transmitted is formed into packets of data bits. If used, forward error correction (FEC) coding is applied to these bits, resulting in binary symbols. For the systems using FEC coding, each packet includes a known header and trailer to allow for decoder synchronization. Each symbol is multiplied by a spreading code, resulting in a number of binary chips. N is the spreading factor in chips per symbol, and W is the total bandwidth expansion due to both FEC coding and direct sequence spreading, in chips per bit. Finally, the chips are input to a modulator which modulates an RF carrier. Thermal noise and a number of interfering signals are added to the transmitted carrier, resulting in a corrupted received signal. At the start of reception, the packet must be acquired by the destination radio. The acquisition process consists of detecting the start of the packet and synchronizing the local spreading code and, for coherent modulation formats, synchronizing to the the carrier. If the packet is successfully acquired, the receiver tracks the spreading code, despreads the signal by multiplying by the code, and demodulates the resulting narrowband signal. The receiver then integrates over one symbol, and makes a hard decision on the symbol value. The decoder estimates the source's data bit stream from the corrupted symbol stream received. If any bit is incorrectly estimated, the entire packet is discarded.

A packet that is generated by a terminal is stored in the PRU in an output packet

buffer. Even after the packet is transmitted, it is retained in this packet buffer until an acknowledgment is received from the destination. A single error (following FEC decoding) in a received packet will destroy the entire packet. Packets include a CRC or parity check code for detection of errors. If a positive acknowledgment is not received, it is assumed that the transmission was unsuccessful, and the packet will be retransmitted. No distinction is made between newly generated packets and those that were previously transmitted without success.

The channel access protocol is disciplined ALOHA [27],[5]. With this protocol, each radio generates potential transmission times, known as the scheduling points, according to a random process, the scheduling process. A Poisson process is chosen for this random scheduling process. These scheduling points dictate the only times at which a radio might begin transmission; however, an actual transmission will begin only if the radio is not already transmitting or receiving a packet from another radio at the time of the scheduling point, and if the radio has a packet available for transmission. Thus, a radio will not interrupt a partially received packet in order to begin a packet transmission.

Each packet has a fixed length preamble for spread spectrum code synchronization. Different spreading codes are used for the preamble and the data portion, and no code is used as both a preamble code and a data code. Both space homogeneous and receiver directed code assignment for the preambles are considered. In the former, all packets use the same code for the preamble; in the latter, each radio has a unique code known by all other radios, and the preamble of each packet uses the code of the destination radio. For space homogeneous preamble codes, addressing information immediately follows the synchronization preamble. The channel access protocol does allow a radio to begin transmitting while it is still attempting to acquire the preamble of a received packet.

The codes used are statistically random sequences, and the sequence is not repeated during the transmission of a packet. This is an accurate representation of a system in which spreading is used primarily for security. It is also an approximation of a system which uses more structured codes, e.g., Gold sequences or Kasami sequences. The approximation is valid for a system with the following structure. Bit-by-bit code changing is used for both the preamble and the data portion of the packets (symbol-by-symbol for channels with FEC coding). The code period is at least one bit. The number of available codes is much larger than the average number of bits (symbols) per packet, so it is very unlikely that the code sequence will have to be repeated during the transmission of a single packet.

When considering the reception of a given packet, the probability that an interfering signal is using the same code as the desired signal depends upon the code assignment policy for the data portion. However, with bit-by-bit code changing, even in the worst case of space homogeneous code assignment, in which all radios transmit using the same sequence of codes, a packet will suffer interference on the same code only if another packet transmission is started within a time interval of plus or minus one bit. If the mean packet length is 500 bits or more and the traffic level is not extremely high, this event is very unlikely. Thus, the code assignment for the data portion can be considered to be either transmitter directed, receiver directed, or space homogeneous, since there will be only a small difference in performance.

2.2 Modeling Assumptions

In order to derive a model which leads to a tractable solution, a number of assumptions and approximations are necessary. These are described below, with a

discussion of the motivation for and validity of some of the modeling choices. The simplest model presented in Chapter 4 is an infinite population model, in which the aggregate traffic is considered. A more exact finite population model is also presented, in which the network is made up of M identical users. Most of the assumptions apply to both infinite and finite population networks.

The overhead needed for FEC decoder synchronization is considered to be negligible, as is the overhead for CRC or parity bits used for error detection. The effect of acknowledgments is ignored; it is assumed that a perfect and instantaneous acknowledgement channel is available. In the case of space homogeneous preamble codes, the required addressing information immediately follows the preamble, and is instantaneous and error free. Consequently, at the beginning of the data portion, only the intended destination will continue to receive the packet, and no packets are lost due to addressing errors. Chip, symbol, and carrier tracking is perfect, and error detection does not occur until the end of a packet. As a result, if the destination acquires a packet, the receiver remains locked onto the packet until the end of transmission, regardless of whether or not bit errors occur.

To reduce the number of variables that must be considered, it is assumed that all users are identical. Therefore, all source-destination pairs have equal traffic requirements, equal scheduling process rates and identical packet length distributions. Also, every radio receives equal power signals from every other radio. In general, zero propagation delay is assumed, so the network state as seen by every radio is identical. The one exception is the approximate treatment of propagation delay in the channel load sense model.

The analysis yields the upper limit on throughput that is asymptotically approached as the delay goes to infinity. A system with infinite delay can be modeled in the following manner. The output packet buffer has infinite capacity, and is full.

Packets to be transmitted are selected randomly from this buffer. Thus, the lengths of successive packets transmitted by a PRU are independent, and there is a packet available for transmission at every scheduling point. Also, packets are never lost at the generating PRU due to buffer overflow. As a result of these assumptions, it is not necessary to distinguish between radios that are backlogged and those that are not.

A Markovian model requires that the transmitted packet lengths be drawn independently from an exponential distribution, and that transmission times be drawn from a Poisson process. For this reason, a Poisson process is chosen for the scheduling process. For the infinite population model, the aggregate rate of the scheduling processes for all users is denoted by Λ. For the finite population models, the rate per user is denoted by λ.

The exponential distribution is not representative of packet lengths found in actual radio networks. However, results from simulation of a system with fixed length packets were very similar to the analytical results derived here. Thus, even though the exact performance will depend upon the packet length distribution, the relative performance as a function of the various channel level and network level parameters is not strongly dependent upon this distribution. Because the Markovian model allows many other details of the network to be accounted for, the approximation of an exponential distribution is used. The average length of a transmitted packet is $1/\mu$. The normalized rate of the scheduling process is referred to as the traffic rate, denoted by $G = \Lambda/\mu$ or by $g = \lambda/\mu$.

The exponential distribution is the distribution of the lengths of transmitted packets, which includes both new packets and retransmissions. But, it is more likely that there is a transmission error in a long packet than in a short one, so the probability of success for a short packet is greater than that for a long packet.

Therefore, the distribution of lengths of successful packets, which are the packets supplied by the terminals, is skewed from this exponential. Also, the length of the packet includes both the data portion and the preamble portion. Yet, the preamble is of fixed length, which leads to the anomaly that for some packets drawn from the exponential distribution, the total length may be shorter than the preamble, which could not occur in a real system. Nevertheless, if the preamble is much less than $1/\mu$, almost all of the packets will be longer than the preamble, so the inaccuracy of the model will have a negligible effect on the results. This has been verified by a simulation model, which specifically accounts for the fixed length preamble.

2.3 Summary

In the first section of this chapter, the system being studied was described in detail. Several restrictions were introduced, such as uniformity among all radios, which will simplify the task of analysis. The second section discussed the approximations that were needed to lead to an analytically tractable solution. The resulting model will be analyzed in Chapter 4, leading to the throughput and probability of success.

Chapter 3

Spread Spectrum Radio Channel

In the literature, the bit error rate (BER) for a code division multiple access (CDMA) channel has been derived for a variety of modulation formats. This chapter consists of a review of one model of an SSMA radio channel, and also a summary of several known results. The mathematical model used was described by Pursley in [38]. When random coding is used, as is assumed in the model of the spread spectrum multiple access (SSMA) channel, a simple approximation to the mean probability of bit error, or bit error rate (BER), can be found. The resulting approximate BER's for several types of modulation are listed. Results are given for both synchronous systems, in which all of the received signals are chip aligned, and for the more applicable case of an asynchronous system, in which the chip offests are random. A bound on the BER for an asynchronous BPSK system, which was described by Taipale in [50], is also presented.

3.1 SSMA System Considerations

The three basic spread spectrum techniques are time hopping (TH), frequency

hopping (FH) and direct sequence (DS). In this dissertation, only direct sequence spreading is considered. The spreading in a direct sequence system stems from the relationship that bandwidth is proportional to signalling rate. The data sequence is multiplied by a high rate sequence (the code sequence), resulting in a chip sequence. The bandwidth of the transmitted signal is proportional to the rate of the chip sequence. The ratio of the chip rate to the data rate, denoted by W, is also the amount by which the signal bandwidth expands. The receiver multiplies the incoming signal by a local version of the code sequence. If the local replica is aligned with the code used for transmission, the receiver can reproduce the narrowband data sequence, which can then be demodulated.

In an SSMA system, the received signal consists of the desired signal, interfering signals due to other transmissions, and noise. It will be shown in the next chapter that the time between network transitions is on the order of or greater than 30 bits. Thus, the number of radios transmitting, $X(t)$, changes slowly compared to T_{bit}, the duration of a bit. Accordingly, with high probability the interference $X(t) - 1$ is constant over T_{bit}, and with very high probability changes by no more than 1. Therefore, it will be assumed that this interference is constant during a single bit. With this approximation, the channel can be characterized by a sequence of states X_j, corresponding to $X(t)$ at each bit j of a packet. Furthermore, the probability of error for a given constant number of transmissions X is a sufficient characterization of the channel.

The channel model reviewed in §3.2 yields approximations to or upper bounds on $P_{e,bit}(X)$, the mean probability of bit error for a bit which is received in the presence of $X - 1$ interfering transmissions and Gaussian noise. The model does not account for fading, multipath, or jamming. The receiver is assumed to be perfectly synchronized with the transmitted signal, which includes chip and bit

timing. For the synchronous systems, it is assumed that the propagation delay is negligible compared to the duration of a chip, so that chip alignment can be maintained among all the transmitters. For coherent modulation formats, it is additionally assumed that the carrier phase offset of the desired signal is zero. The received signal power is identical for all transmitter-receiver pairs, and the thermal noise level is identical for all users, so all receptions have the same E_b/N_0, where E_b/N_0 is the ratio of received bit energy to thermal noise density.

As discussed in §2.1, the probability that two transmissions begin within the duration of one bit is negligible. Therefore, it can safely be assumed that all interfering transmissions are using different codes, so the interference is due to cross-correlation between codes, not to the auto-correlation of a given code. In order to avoid calculations of the cross-correlation properties of the specific spreading codes used, the pseudo-noise (PN) codes are modeled as sequences of jointly independent Bernoulli (1/2) random variables. A well chosen set of codes can have lower cross-correlations and consequently lower $P_{e,bit}(X)$ than the random codes used in the model.

In an SSMA system, even if FEC coding is not used, the bit errors are correlated. Because the correlation is positive, the probability that a bit is not in error given that all previous bits were not in error is greater than $1 - P_{e,bit}$. For a packet of length $\mathcal{L}$ bits, the probability of no errors, which is the probability of successful reception, is bounded by

$$P_S \geq \prod_{j=1}^{\mathcal{L}} (1 - P_{e,bit}(X_j)), \tag{3.1}$$

where $P_{e,bit}(X_j)$ is the unconditional probability of an error at the jth bit, which is a function only of X_j. If this bound is used, the channel can be characterized by the function $P_{e,bit}(X)$. This function is the BER of the multiple access channel, and has been analyzed extensively.

3.2 DS-SSMA Radio Channel Model

The model of the DS-SSMA radio channel is one that was described by Pursley in [38]. The received signal $r(t)$ is modeled as the sum of $X(t)$ signals and thermal noise. The signals are indexed by k, with $k = 1$ denoting the desired signal and the other $X - 1$ signals constituting the interference. The spreading code sequence used by the kth radio is $a_{(k,j)}$. After multiplication by the data, the chip sequence is $\tilde{a}_{(k,j)}$. The waveform is denoted by $a_k(t)$, where

$$a_k(t) = a_{(k,j)} \qquad \text{for} \qquad jT_c \leq t < (j+1)T_c, \tag{3.2}$$

and similarly for $\tilde{a}_k(t)$. The channel errors are symmetric, meaning that the probability of error is the same if the transmitted bit is a 1 or a 0. Therefore, $P_{e,bit}$ is found for a transmitted bit of $+1$. The thermal noise is assumed to be white Gaussian noise.

Although it is assumed that the propagation delay is negligible on the time scale of the mean packet length, the delay may be an appreciable fraction of or greater than the duration of a chip, T_c. Also, in the general case of an asynchronous system, each user has a starting time offset. These effects are lumped into a single parameter τ_k, which is the total time offset at the receiver for signal k. Moreover, each signal has an initial carrier phase offset, and an additional offset due to propagation delay. These are accounted for by the parameter ϕ_k, which is a random variable, uniformly distributed between 0 and 2π. The received signal can be expressed as

$$r(t) = n(t) + \sum_{k=1}^{X} \sqrt{2S}\tilde{a}_k(t - \tau_k)\cos(\omega_c t + \phi_k), \tag{3.3}$$

where S is the signal power, $\omega_c/2\pi$ is the carrier frequency, and $n(t)$ is the thermal noise. It is assumed that the carrier and code tracking are perfect, so the offsets τ_1

and ϕ_1 are both zero. The received signal is multiplied by the carrier $\sqrt{2}\cos(\omega_c t)$ and by the spreading code $a_k(t)$, and integrated over one bit, yielding

$$\mathcal{Z} = \int_0^{T_{bit}} r(t)a_1(t)\sqrt{2}\cos(\omega_c t)dt. \tag{3.4}$$

In the more general case of an asynchronous system, the interference can be expressed in terms of the aperiodic partial cross-correlation between the respective codes. The time offset τ_k can be written as the sum of the integer and fractional parts,

$$\tau_k = J_k + \delta_k; \qquad 0 \le \delta_k \le 1. \tag{3.5}$$

The fractional part δ_k is uniformly distributed between 0 and 1. As shown in Fig. 3.1, the cross-correlation between $a_1(t)$ and $\tilde{a}_k(t-\tau_k)$ is equal to

$$A_k = \frac{1}{N}\sum_{j=1}^{N}(1-\delta_k)\tilde{a}_{(k,j')}a_{(1,j)} + \delta_k\tilde{a}_{(k,j'-1)}a_{(1,j)}; \qquad j - j' = J_k. \tag{3.6}$$

Neglecting the double frequency term, $\mathcal{Z}$ is equal to

$$\begin{aligned}\mathcal{Z} &= T_{bit}\sqrt{S}\left(1 + \sum_{k=2}^{X} A_k\cos(\phi_k)\right) + \mathcal{N} \\ &= T_{bit}\sqrt{S}(1+I) + \mathcal{N},\end{aligned} \tag{3.7}$$

where the thermal noise $\mathcal{N}$ is a zero-mean Gaussian r.v. with variance $T_{bit}N_0/2$, and I is the normalized interference term.

With random coding, the variable A_k is the weighted sum of two independent binomial random variables, $A_k = (1-\delta_k)A_k^0 + \delta_k A_k^1$, where A_k^0 and A_k^1 are each the sum of N independent Bernoulli random variables. The Bernoulli r.v.'s take values of $1/N$ and $-1/N$, each with probability $1/2$. Therefore, the means of A_k^0

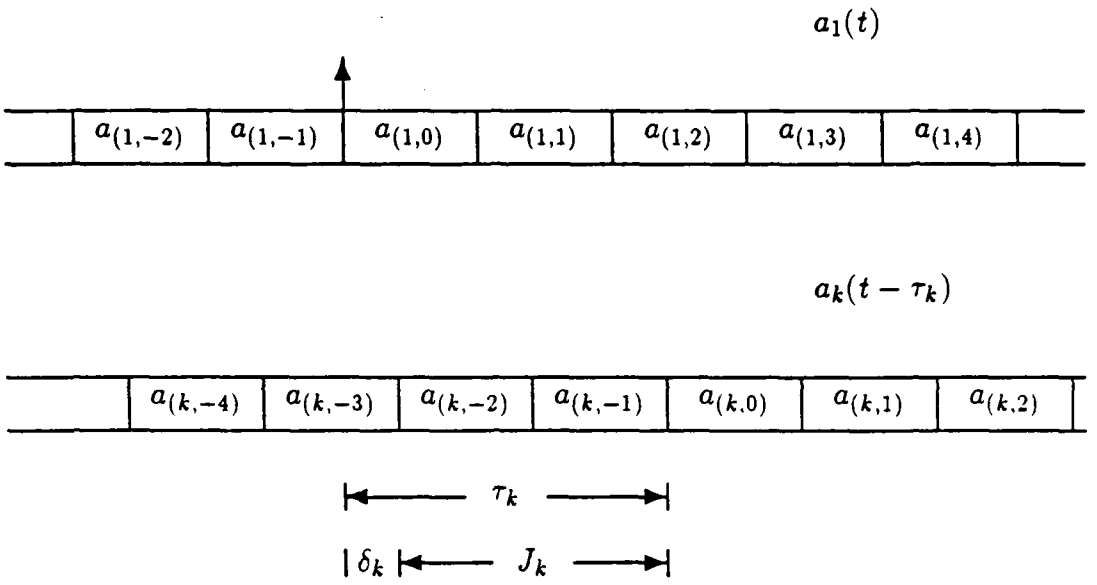


Fig. 3.1 Cross-correlation with interfering signal.

and A_k^1 are both zero, and the variances, $1/N$. This model of the SSMA radio channel is used in the next two sections to find approximations to and a bound on the probability of bit error.

3.3 Approximations to the Probability of Bit Error

In this section, an approximation to the probability of bit error is derived for the asynchronous BPSK SSMA radio channel. The approximation is valid for large spreading factors N and for many simultaneous transmission X. For even moderate values of N, A_k^0 and A_k^1 can be closely approximated as zero mean Gaussian r.v.'s, with variance $1/N$. With this approximation, for $k \neq 1$, the sum A_k of Eqn. 3.6 is approximately Gaussian, with zero mean and variance $(1 - 2\delta_k + 2\delta_k^2)/N$. Furthermore, the total normalized interference I is also a zero mean Gaussian random variable, with variance

$$\begin{aligned} \sigma^2 &= \sum_{k=2}^{X} \sigma_k^2 \\ &= \sum_{k=2}^{X} \frac{(1 - 2\delta_k + 2\delta_k^2)}{N} \cos^2(\phi_k). \end{aligned} \tag{3.8}$$

Thus, assuming that the terms A_k are Gaussian, I is the sum of many Gaussian random variables, each of which has a variance σ_k^2 that depends upon the offsets δ_k and ϕ_k. The probability of error given X is found by averaging $P_{e,bit}$ over all offsets. However, this averaging requires a great deal of computation. As X becomes large, the variance of I will approach $X - 1$ times $E(\sigma_k^2)$, the expected value of σ_k^2 averaged over δ_k and ϕ_k. Therefore, an approximation to $P_{e,bit}$ valid for large X is found by treating I as a Gaussian random variable with a variance of $(X-1)E(\sigma_k^2)$. The average over all values of δ_k of σ_k^2 is $(2/3N)\cos^2(\phi_k)$, so $E(\sigma_k^2) =$

$1/3N$. This approximation is valid only for large values of X. However, for small X, the probability of error is very small, so the inaccuracy of the approximation in this region has little effect on the overall performance.

With these approximations, $\mathcal{Z}$ is the sum of the desired signal plus two Gaussian r.v.'s,

$$\mathcal{Z} = T_{bit}\sqrt{S} + T_{bit}\sqrt{S}I + \mathcal{N}, \tag{3.9}$$

where I is zero-mean, with variance $(X-1)/3N$. Because $I + \mathcal{N}$ is also Gaussian, the probability of error can be found from standard communications theory.

$$P_{e,bit}(X) = Q(SNR_e). \tag{3.10}$$

where $Q(\xi)$ is the complement of the c.d.f. of a normal distribution,

$$Q(\xi) = \frac{1}{\sqrt{2\pi}} \int_{\xi}^{\infty} e^{-\gamma^2/2}\, d\gamma, \tag{3.11}$$

SNR_e is the effective signal to noise ratio,

$$\frac{1}{SNR_e} = \sqrt{\frac{N_0}{2E_b} + \frac{(X-1)}{3N}}, \tag{3.12}$$

and $E_b = ST_{bit}$.

A similar analysis leads to the probability of bit error for various other modulation formats. The systems are classified as either synchronous, in which the chip offsets δ_k are all zero, and asynchronous, in which the δ_k are uniformly distributed in $[0, 1]$. For most packet radio networks, the propagation delay will be large, so synchronous operation will not be feasible. Nevertheless, for comparison purposes, results are given for synchronous channels as well as for the more applicable asynchronous channels. The systems are also classified as coherent, in which the carrier

is tracked so the phase offset $\phi_1 = 0$, and non-coherent, in which the phase offset is non-zero, and may vary over time. Results for the coherent systems are summarized by Geraniotis in [20]. $P_{e,bit}(X)$ has the form given by Eqn. 3.10. The effective signal to noise ratios are listed below.

Asynchronous BPSK:

$$\frac{1}{SNR_e} = \sqrt{\frac{N_0}{2E_b} + \frac{(X-1)}{3N}} \tag{3.13}$$

Asynchronous QPSK:

$$\frac{1}{SNR_e} = \sqrt{\frac{N_0}{2E_b} + \frac{2(X-1)}{3N}} \tag{3.14}$$

Synchronous BPSK:

$$\frac{1}{SNR_e} = \sqrt{\frac{N_0}{2E_b} + \frac{(X-1)}{2N}} \tag{3.15}$$

Synchronous QPSK:

$$\frac{1}{SNR_e} = \sqrt{\frac{N_0}{2E_b} + \frac{(X-1)}{N}} \tag{3.16}$$

Geraniotis has also found the probability of error for several non-coherent systems [GERAb]. The two formats considered here are differential phase shift keying (DPSK) and binary frequency shift keying (FSK), for which $P_{e,bit}(X)$ has the form

$$P_{e,bit}(X) = 1/2\epsilon^{-\alpha}. \tag{3.17}$$

The term α is found from

Asynchronous DPSK:

$$\frac{1}{\alpha} = -2\left(\frac{N_0}{2E_b} + \frac{(X-1)}{3N}\right) \tag{3.18}$$

Asynchronous FSK:

$$\frac{1}{\alpha} = -4\left(\frac{N_0}{2E_b} + \frac{(X-1)}{6N}\right) \tag{3.19}$$

Synchronous DPSK:

$$\frac{1}{\alpha} = -2\left(\frac{N_0}{2E_b} + \frac{(X-1)}{2N}\right) \tag{3.20}$$

Synchronous FSK:

$$\frac{1}{\alpha} = -4\left(\frac{N_0}{2E_b} + \frac{(X-1)}{4N}\right) \tag{3.21}$$

3.4 Bound on the Probability of Bit Error

As stated previously, even when conditioned on X_j, the channel errors are correlated through the dependence on ϕ_k and δ_k. Furthermore, the derivations are approximate, relying on the average values of these offsets. An upper bound on $P_{e,bit}$ can be derived by considering the worst case for the offsets. Pursley has shown that the worst case offsets are $\phi_k = 0$ and $\delta_k = 0$ [42]. The model resulting from this worst case was described by Taipale in [50]. The modulation format is coherent BPSK. Although the model is actually a synchronous channel, it leads to a bound on the performance of the asynchronous channel.

For a system with bit-by-bit code changing, with this worst case assumption, the codes used for the interfering transmissions change exactly at the bit boundaries of the desired signal. Therefore, the effect of the interference, and consequently the probability of error, is independent from bit to bit. Because of this independence, the probability of bit error for an individual bit is only a function of X, and this function $P_{e,bit}(X)$ is a sufficient characterization of the channel. The conditional independence of errors is necessary in the derivation of the model of the Viterbi decoder in Chapter 5.

With the offsets equal to zero, the terms A_k are binomial random variables, equal to the sum of N Bernoulli r.v.'s which are $1/N$ and $-1/N$ with probabilities $1/2$ and $1/2$. Furthermore, the normalized interference I is the sum of $X-1$ such A_k, so is itself a shifted binomial distribution, with a zero mean, a maximum of $(X-1)$, and a variance of $(X-1)/N$. From Eqn. 3.7, conditioned on I, the output of the receiver is a Gaussian r.v., mean $T_{bit}\sqrt{S}(1+I)$ and variance $T_{bit}N_0/2$. The average probability of bit error can be found by averaging over all values of I, so

$$P_{e,bit}(X) \leq \sum_{i=0}^{N(X-1)} \binom{N(X-1)}{i} 2^{-N(X-1)} p(i), \tag{3.22}$$

where

$$p(i) = Q\left(\sqrt{2E_b/N_0}\left(1+\frac{2i-N(X-1)}{N}\right)\right). \tag{3.23}$$

An inequality is used because the expression found is a bound on the probability of bit error for an asynchronous system.

Using a Gaussian approximation for the binomial distribution of interfering chips gives

$$P_{e,bit}(X) \lesssim Q(SNR_e), \tag{3.24}$$

where

$$\frac{1}{SNR_e} = \sqrt{\frac{N_0}{2E_b} + \frac{(X-1)}{N}}. \tag{3.25}$$

Comparing to the approximation of Eqn. 3.15, this is seen to be off by a factor of 1/2 in the interference term. This factor stems from the averaging over all carrier offsets in the approximation. Coincidentally, this approximation is equal to the approximation for the synchronous QPSK channel.

The approximation of Eqn. 3.24 is found to be accurate to within 1% of the results from Eqn. 3.22 when $N(X-1)$ is greater than 512 and $P_{e,bit}(X) > 10^{-10}$. For very small $P_{e,bit}(X)$, the relative accuracy becomes poor. However, error rates this low have virtually no impact on the network performance. For large values of $N(X-1)$, the approximation requires significantly less computing time than the exact expression. Therefore, $P_{e,bit}(X)$ is calculated using Eqn. 3.22 when $N(X-1) \leq 512$ and using Eqn. 3.24 when $N(X-1) > 512$.

3.5 Summary

In this chapter, a model of a DS-SSMA radio channel was reviewed. Several known approximations and bounds on the probability of bit error $P_{e,bit}(X)$ were listed. The approximations that were summarized in this chapter will be used in Chapter 4 to find the throughput of an SSMA PR network. Most of the network results given in later chapters will make use of the bound derived in §3.4. Because this bound leads to errors which are conditionally independent given X, it can be used as the channel model for a system using FEC coding, as discussed in Chapter 5. In order to give a fair comparison between coded and uncoded channels, this bound is also used for many of the uncoded channel results.

Chapter 4

Throughput Analysis

In this chapter, the network throughput and the probability of success are determined for the SSMA PR network model described in Chapter 2. The model incorporates the detailed radio channel of Chapter 3, and allows the throughput to be found as a function of the probabilities of bit error, $P_{e,bit}(X)$, which are in turn a function of the channel parameters. Numerical results are found for the uncoded radio channel, with the assumption that the synchronization process is ideal. These results show the throughput as a function of the received signal power E_b/N_0, the spreading factor W, the traffic rate g, and the network size M. Also, several different modulation formats are compared.

The network model specifically accounts for the varying level of interference encountered during a packet reception. It takes as parameters the probabilities of bit error for a given constant level of interference, $P_{e,bit}(X)$. Any radio channel can be incorporated, provided that the channel can be modeled as memoryless, meaning that the probability of error at a given bit position depends upon the network evolution only through the current level of interference X. Most radio channels without FEC coding are memoryless. Furthermore, in the next chapter,

an approximate model of a channel with FEC coding is shown to be memoryless.

In addition to the channel errors, the network model takes as an input the probability that a packet is received, which depends upon the performance of the synchronization circuit. Consequently, this same network model can be used either with the idealized model that the synchronization process is perfect, or with the accurate model of the synchronization process which is developed in Chapter 6. The general model of the SSMA PR network analyzed in this chapter is flexible, and can be used in conjunction with a variety of channel models and models of the synchronization process.

There are several levels of complexity for the network model. The simplest model, presented in §4.1, makes use of an infinite population assumption. This model shows the effect of channel errors on network throughput. A more accurate finite population model is presented in §4.2. In an actual network, there may be an appreciable chance that a packet will not be received because the destination is busy. This effect is referred to as receiver availability. The finite population model takes this effect into account. This model is then further extended in §4.3 to examine the performance of a system which uses a channel load sensing protocol to limit the number of transmissions. An approximate model of a system with non-negligible propagation delay is also developed. Numerical results show the improvement in network throughput due to channel load sensing.

4.1 Infinite Population Model

The performance of a very large network can be approximated by assuming that the network consists of an infinite number of radios. For such a model, the aggregate traffic on the channel can be modeled as generated by a Poisson process with rate

Λ. As described in §2.2, the packets are assumed to be exponentially distributed with mean $1/\mu$. The traffic rate is $G = \Lambda/\mu$. With these assumptions, the network can be modeled by a continuous time Markov chain. The network state $Z(t)$ is simply the number of radios transmitting at time t, denoted by $X(t)$. The state space Ω of $Z(t)$ is $\Omega = \{0, 1, 2, ...\}$. This *primary Markov chain* is irreducible and homogeneous, so a strictly positive stationary distribution $\{\pi_0(X),\ X \in \Omega\}$ can be found. This system is equivalent to a queue with an infinite number of servers, and the state transition rates are same as those of the $M/M/\infty$ queue [KLEI75], shown in Fig. 4.1. The transition rate matrix, denoted by $\mathbf{Q}$, is a simple band matrix. The steady state probability distribution is

$$\pi_0(X) = \frac{(\Lambda/\mu)^X}{X!} e^{(-\Lambda/\mu)} \qquad X = 0, 1, 2, ... \tag{4.1}$$

4.1.1 Network Throughput

For the infinite population model, the network throughput S is defined as

$$S \triangleq \lim_{T \to \infty} \frac{1}{T} (\text{Total length of packets successfully transmitted in interval } T). \tag{4.2}$$

A common technique used for evaluating throughput is to find the probability of success for a packet and multiply this by the average length of successful packets. For systems with fixed length packets, and also for perfect capture systems, the average length of successful packets is equal to the average length of transmitted packets. However, in the system being considered, the probability of success depends on the packet length, so the average length of successful packets is not equal to $1/\mu$. Therefore, a more involved derivation of throughput is required. The method used was first found by Brazio in [5].

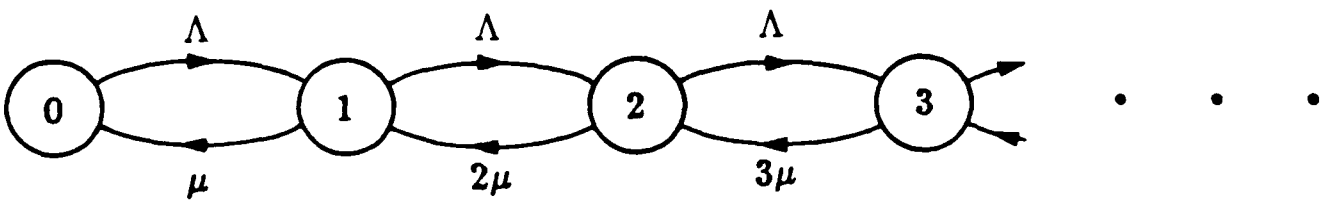


Fig. 4.1 State-transition-rate diagram for the infinite population model.

T_s, the contribution to throughput of a single packet, is defined to be the length of the packet if it is successful, and zero if unsuccessful. $\overline{T}_s(X)$ denotes the expected contribution to throughput of a packet given that the network is in state X just prior to the start of transmission. This expectation is over all future evolutions of the network, which takes into account all combinations of bit error rates that might be encountered by a packet starting in X. It should be emphasized that the evaluation of $\overline{T}_s(X)$ does not assume that the network state is X throughout the entire packet. The probability of success of a packet of length τ given the same conditions is denoted by $P_{S|X,\tau}$ and the average over all lengths τ by $P_{S|X}$.

The synchronization process is assumed to be perfect, so the probability of success is given by

$$P_S = \sum_{X=0}^{\infty} \Lambda \pi_0(X) P_{S|X}, \tag{4.3}$$

and the throughput is given by

$$S = \sum_{X=0}^{\infty} \Lambda \pi_0(X) \overline{T}_s(X). \tag{4.4}$$

4.1.1.1 Auxiliary Markov Chain

The expected contribution to throughput, $\overline{T}_s(X)$, is found by considering the states of the network during the reception of a specific packet, referred to as the tagged packet. The evolution of the network during the reception of the tagged packet can be represented by a Markov chain, which is called the *auxiliary Markov chain*, $Z_{\text{aux}}(t)$. The state space of $Z_{\text{aux}}(t)$, denoted by Ω_{aux}, consists of all of the states of the primary Markov chain that could be visited by the tagged packet, and two absorbing states, *Success* and *Failure*. The subset of Ω_{aux} made up of non-absorbing states is denoted by Ω^*_{aux}. For the uncoded channel, the state *Failure*

is entered when a channel error occurs. For the channel with FEC coding, the state *Failure* is entered when a decoder error occurs. If the packet is completed without entering the state *Failure*, the transmission is successful, so the state *Success* is entered.

As discussed in Chapter 3, for a system that does not use FEC coding, the probability of success is lower bounded by considering the channel errors to be conditionally independent from bit to bit, given the number of transmitters X_j. The bit error process can be modeled as a time-varying Bernoulli process, in which the probability of bit error at bit j depends upon X_j. The occurrence of decoder errors from the decoder model derived in Chapter 5 is also shown to be a time-varying Bernoulli process, where the probability of decoder error is a function only of the current number of transmissions. For both cases, the duration of a single bit is small compared to the time between network transitions, so at the network level the occurrence of errors can be closely approximated as a continuous process. This continuous time approximation is a time-varying Poisson process, at a rate $\epsilon(X(t))$ which is a function of the number of transmissions $X(t)$. The rates of the Poisson process, $\epsilon(X)$, are found by setting the probability of no arrivals in an interval of length T_{bit} equal to $1 - P_{e,bit}(X)$. In the next chapter, it will be shown that for the coded channel, $1 - P_{e,bit}(X)$ is replaced by $P_C(P_{e,symbol}(X))$, the probability of a correct decoder decision given a probability of symbol error $P_{e,symbol}(X)$. In the case of the uncoded channel,

$$e^{-\epsilon(X)T_{bit}} = 1 - P_{e,bit}(X);$$

$$\epsilon(X) = \frac{-\ln\left(1 - P_{e,bit}(X)\right)}{T_{bit}}. \tag{4.5}$$

Normalizing to the mean packet length,

$$\frac{\epsilon(X)}{\mu} = \frac{-\ln\left(1 - P_{e,bit}(X)\right)}{\mu T_{bit}}. \tag{4.6}$$

The quantity $1/\mu T_{bit}$ is equal to the mean packet length divided by the duration of a bit, which is the average number of bits per packet. Thus,

$$\frac{\epsilon(X)}{\mu} = -\ln\left(1 - P_{e,bit}(X)\right) b, \tag{4.7}$$

where b is the average number of bits per packet.

Because the process is Poisson, the time until the occurrence of an error is exponentially distributed. Thus, the occurrence of channel errors, or decoder errors in the case of FEC coding, can be modeled as transitions to the state *Failure* from each state X at rate $\epsilon(X)$. Also, the time to completion of a given packet is exponentially distributed with rate μ, independent of the network state. This is modeled as transitions from every non-absorbing state to the absorbing state *Success* at rate μ.

As stated so far, the auxiliary Markov chain consists of an infinite number of states. However, for any realistic channel, as the number of transmissions increases, the probability of error increases to the limiting value of 0.5. In addition, if the probability of error is 0.5 for even 10 bits, the probability that the packet is successful will be less than 0.1%. Therefore, a cutoff L can be found such that if $X > L$, then the probability that the packet is successful is negligible. With the cutoff L, the state space Ω_{aux} consists of $\Omega^*_{\text{aux}} = \{X | 1 \leq X \leq L\}$ and the states *Success* and *Failure*, as shown in Fig. 4.2.

Indexing *Success* as state $L+1$ and *Failure* as $L+2$, the transition rate matrix

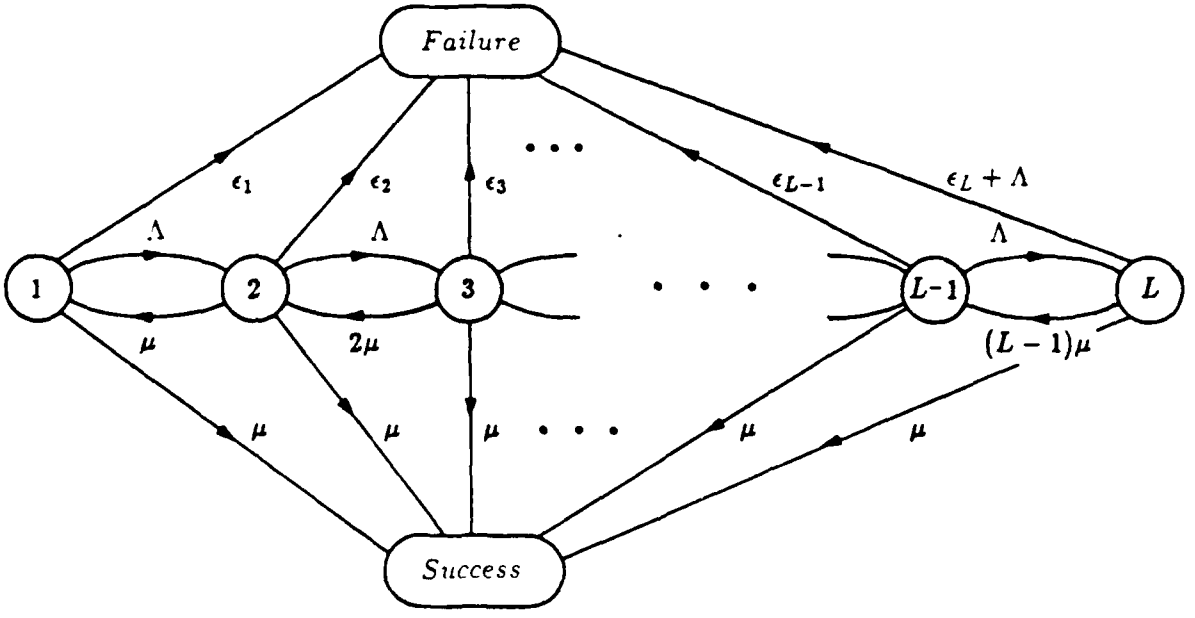


Fig. 4.2 Auxiliary Markov chain for the infinite population model.

$\mathbf{Q}_{\text{aux}}$ is

$$\mathbf{Q}_{\text{aux}} = \begin{pmatrix} \mathbf{Q}^*_{\text{aux}} & \mu\underline{1}^T & \underline{\varphi}^T \\ \underline{0} & 0 & 0 \\ \underline{0} & 0 & 0 \end{pmatrix}, \tag{4.8}$$

where $\mathbf{Q}^*_{\text{aux}}$ is the $L \times L$ sub-matrix corresponding to transitions between the states $\{1, 2, ..., L\}$, $\underline{\varphi}^T$ is the $L \times 1$ vector corresponding to transitions to the state *Failure*, $\underline{1}^T$ is the $L \times 1$ vector of ones, and $\underline{0}$ is the $1 \times L$ vector of zeros. The elements of $\mathbf{Q}^*_{\text{aux}}$ can be derived from the transition rate matrix $\mathbf{Q}$ of the original Markov chain and the values of $\epsilon(X)$, as

$$[\mathbf{Q}^*_{\text{aux}}]_{(i,j)} = \begin{cases} (i-1)\mu & j = i-1; \\ -(i\mu + \Lambda + \epsilon(i)) & j = i; \\ \Lambda & j = i+1 \end{cases} \qquad \begin{array}{c} 1 \le i \le L \text{ and} \\ 1 \le j \le L \end{array} \tag{4.9}$$

4.1.1.2 Calculation of $\overline{T}_s(X)$ and $P_{S|X}$

$\overline{T}_s(X)$ and $P_{S|X}$ can be found from the transition rate matrix of the auxiliary Markov chain $\mathbf{Q}_{\text{aux}}$. First, $\overline{T}_s(X)$ and $P_{S|X}$ are found from $P_{S|X,\tau}$ by removing the condition on the packet length,

$$\begin{aligned} \overline{T}_s(X) &= \int_0^\infty \tau P_{S|X,\tau} f(\tau)\, d\tau; \\ P_{S|X} &= \int_0^\infty P_{S|X,\tau} f(\tau)\, d\tau, \end{aligned} \tag{4.10}$$

where $f(\tau)$ is the p.d.f. of the packet lengths, $f(\tau) = \mu e^{-\mu\tau}$.

The product $f(\tau)P_{S|X,\tau}$ is equal to the rate at which $Z_{\text{aux}}(t)$ enters the state *Success* at time τ given that $Z_{\text{aux}}(0^+) = X + 1$. Conditioning on the state at time τ^- gives

$$f(\tau)P_{S|X,\tau} = \sum_{X' \in \Omega^*_{\text{aux}}} \mu \Pr\Big(Z_{\text{aux}}(\tau^-) = X' | Z_{\text{aux}}(0^+) = X + 1\Big). \tag{4.11}$$

since there is a transition from every non-absorbing state to the state *Success* at a rate μ, independent of τ.

$[\mathbf{A}]_{(j,i)}$ will be used to denote the j,ith element of a matrix $\mathbf{A}$, and $[\underline{v}]_j$ to denote the jth element of a vector $\underline{v}$. For a homogeneous Markov chain with transition rate matrix $\mathbf{Q}$, the probability of being in state i at time τ given starting state j at time 0 is $[e^{\mathbf{Q}\tau}]_{(j,i)}$, where

$$e^{\mathbf{Q}\tau} = I + \mathbf{Q}\tau + \mathbf{Q}^2\tau^2/2! + \mathbf{Q}^3\tau^3/3! + \ldots \tag{4.12}$$

If index j corresponds to the state $X+1$, Eqn. 4.11 becomes

$$f(\tau)P_{S|X,\tau} = \sum_{i=1}^{L} \mu[e^{\mathbf{Q}\mathrm{aux}\tau}]_{(j,i)}. \tag{4.13}$$

This is simplified by noting that only the $L \times L$ upper left corner of $e^{\mathbf{Q}\mathrm{aux}\tau}$ is needed, which is equal to $e^{\mathbf{Q}^*_{\mathrm{aux}}\tau}$. Thus,

$$f(\tau)P_{S|X,\tau} = \mu[e^{\mathbf{Q}^*_{\mathrm{aux}}\tau}\underline{1}^T]_j, \tag{4.14}$$

where $\underline{1}^T$ is the $L \times 1$ vector of ones, so

$$\begin{aligned}
\overline{T}_s(X) &= \int_0^\infty \tau\mu[e^{\mathbf{Q}^*_{\mathrm{aux}}\tau}\underline{1}^T]_j\, d\tau \\
&= \mu[\int_0^\infty \tau e^{\mathbf{Q}^*_{\mathrm{aux}}\tau}\, d\tau \underline{1}^T]_j \\
&= \mu[(\mathbf{Q}^*_{\mathrm{aux}})^{-2}\underline{1}^T]_j.
\end{aligned} \tag{4.15}$$

In a similar fashion,

$$\begin{aligned}
P_{S|X} &= \int_0^\infty \mu[e^{\mathbf{Q}^*_{\mathrm{aux}}\tau}\underline{1}^T]_j\, d\tau \\
&= -\mu[(\mathbf{Q}^*_{\mathrm{aux}})^{-1}\underline{1}^T]_j.
\end{aligned} \tag{4.16}$$

The expression for $\overline{T}_s$ was first derived by Brazio [5] in a different manner by considering the forward Kolmogorov equations for the auxiliary Markov chain.

4.1.1.3 Narrowband Channel

The throughput of the uncoded, unspread radio channel with thermal noise corresponds to $W = 1$ chip per bit, which will give $L = 1$. From Eqn. 4.15,

$$\begin{aligned}\overline{T}_s(0) &= \int_0^\infty e^{-(\Lambda+\epsilon(1))\tau}\, \tau\mu e^{-\mu\tau}\, d\tau \\ &= \frac{1/\mu}{(1+G+\epsilon(1)/\mu)^2}, \end{aligned} \tag{4.17}$$

which gives

$$S = \frac{Ge^{-G}}{(1+G+\epsilon(1)/\mu)^2}. \tag{4.18}$$

If the thermal noise is negligible, $\epsilon(1)/\mu << 1$, so this simplifies to

$$S = \frac{Ge^{-G}}{(1+G)^2}. \tag{4.19}$$

The capacity $C = 0.137$ occurs at $G^* = \sqrt{2} - 1$. This result was previously derived by Ferguson [16] and by Bellini and Borgonovo [2].

4.1.2 Numerical Calculation

The expected contribution to throughput is found numerically by solving

$$(-\mathbf{Q}^*_{\text{aux}}/\mu)\underline{x} = \underline{1}^T \qquad \text{and} \qquad (-\mathbf{Q}^*_{\text{aux}}/\mu)\underline{y} = \underline{x}. \tag{4.20}$$

The linear equations are solved using the SOR routine from the ITPACK package [24]. The routines were changed to double precision to avoid numerical instability. Throughput is found as

$$S = G\sum_{i=1}^{L} [\underline{y}]_i \pi_0(i-1). \tag{4.21}$$

The throughput is a unimodal function of the traffic rate G. This means that an optimum value G^* exists such that $S(G)$, the throughput for a traffic rate G, is monotonically increasing for $0 \leq G < G^*$ and is monotonically decreasing for $G^* < G < \infty$. The maximum $S(G^*)$ is referred to as the capacity C. The capacity is found by using Brent's method for one-dimensional minimization [8]. The function minimized is $-S(G)$, which results in the maximum of $S(G)$. Brent's method requires starting points G^- and G^+ such that $G^- < G^* < G^+$. These points are found by using a modification of the routine MNBRAK from [37]. The routine was modified to make use of the constraint $G^* > 0$, leading to a smaller interval $G^+ - G^-$ when the initial guess $G_0 > G^*$. When finding the capacity as a function of the various parameters, the value G^* found in the last iteration is used as the next starting point for MNBRAK.

For the uncoded channel, the probability of bit error changes slowly with X, so the cutoff L described in §4.1.1.1 is often greater than several hundred. However, the optimum numerical value of G^* is usually much smaller than this L. Because the probability of being in a state $X >> G^*$ is very small, there will be very little error in choosing a smaller value of L, which will require less computing time. Thus, L is chosen as the largest X such that $P_{e,bit}(X) \leq \xi$. The parameter ξ is chosen in the interval $[0.01, 0.5]$. After finding the approximate G^*, the probability of being in state L, $\pi_0(L)$ is checked. If $\pi_0(L) < 0.001$, the approximation is deemed valid, otherwise, a larger value of ξ is chosen.

4.1.3 Infinite Population Results

Throughout this dissertation, all numerical results are found for a mean packet length of $1/\mu = 1000$ bits. The throughput S and probability of success P_S are plotted as a function of the traffic rate G in Fig. 4.3. The bound on $P_{e,bit}$ for

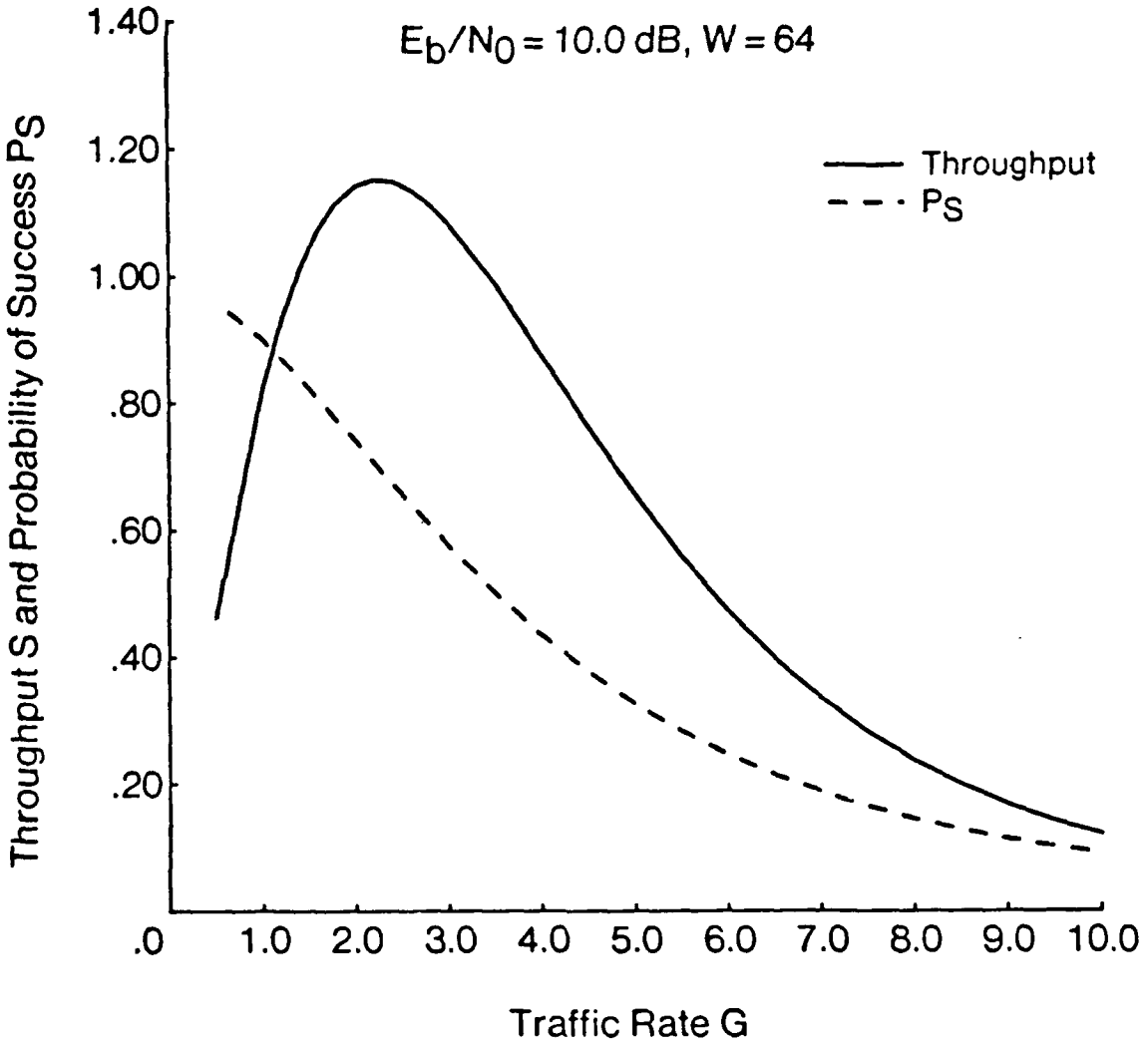


Fig. 4.3 Throughput and probability of success versus traffic rate, infinite population, uncoded channel.

the BPSK channel was used, with an E_b/N_0 of 10.0 dB and a spreading factor of $W = 64$ chips per bit*.

The bandwidth required is proportional to the spreading factor W. Therefore, it is of interest to consider the normalized throughput, S/W. In general, the normalized capacity of the SSMA system is lower than the capacity of the narrowband ALOHA system. However, the capacity of the narrowband system is achieved at a G^* which gives a low probability of success. Pursley [40] has introduced a performance measure which is the maximum throughput achievable with the constraint that P_S exceeds a minimum value. Typically, the constraint on P_S is 0.95 or 0.99. This measure arises from the consideration of real time applications such as packetized voice. For such applications, packets that incur a substantial delay will be discarded. If it is assumed that the sum of the initial queueing delay and the transmission time is smaller than the maximum allowable delay, but that the retransmission delay is larger than this maximum delay, then packets which are successful on the first try will be useful, but packets which must be retransmitted will suffer too much delay, so will be discarded. Therefore, the constraint on P_S gives one minus the allowable fraction of packets which can be discarded.

The maximum normalized throughput S/W with constrained P_S is tabulated in Table 4.1 for an E_b/N_0 of 10.0 dB, for several values of the constraint on P_S, for both the narrowband channel, and the SSMA channel with $W = 64$. Even under this criterion, the normalized uncoded SSMA channel does not compare favorably to the narrowband channel. However, in the next chapter, it will be shown that with the use of FEC coding, the SSMA channel outperforms the narrowband channel under this criterion.

In Fig. 4.4, S/W is plotted as a function of G/W for several values of W. For the

*For the uncoded channel, $W = N$.

Minimum P_S	Narrowband	SSMA
0.90	0.0442	0.0129
0.95	0.0223	0.00864
0.99	0.00307	0.00192

Table 4.1. Maximum throughput with constrained P_S, uncoded narrowband and SSMA channels.

infinite population model, the peak normalized throughput, which is the normalized capacity, increases with higher spreading factors. Although this is often the case, at low E_b/N_0 the normalized capacity can decrease with increasing W.

In Fig. 4.5, the capacity C is plotted as a function of W for several values of E_b/N_0, and for the limiting case of no thermal noise, $E_b/N_0 = \infty$. The capacity increases almost linearly with W, with a higher slope for higher E_b/N_0. Fig. 4.6 is a similar plot for the normalized capacity C/W. Except for low values of E_b/N_0, the normalized capacity increases with increasing W.

A comparison of the coherent modulation formats of §3.3 is shown in Fig. 4.7 for a spreading factor of $W = 64$. As expected, the ranking is the same as the ranking for $P_{e,bit}$, with asynchronous BPSK being the best, followed by synchronous BPSK, asynchronous QPSK, and synchronous QPSK. The throughput using the upper bound on $P_{e,bit}$ of Section §3.4 is almost indistinguishable from the throughput for the synchronous QPSK channel, which corresponds to a Gaussian approximation to the binomial distribution of the interference. The bound is much worse than the approximate performance, given by the highest curve. Nevertheless, results found using the bound in this and later chapters will indicate the trends and sensitivities of the network performance to the parameters under question.

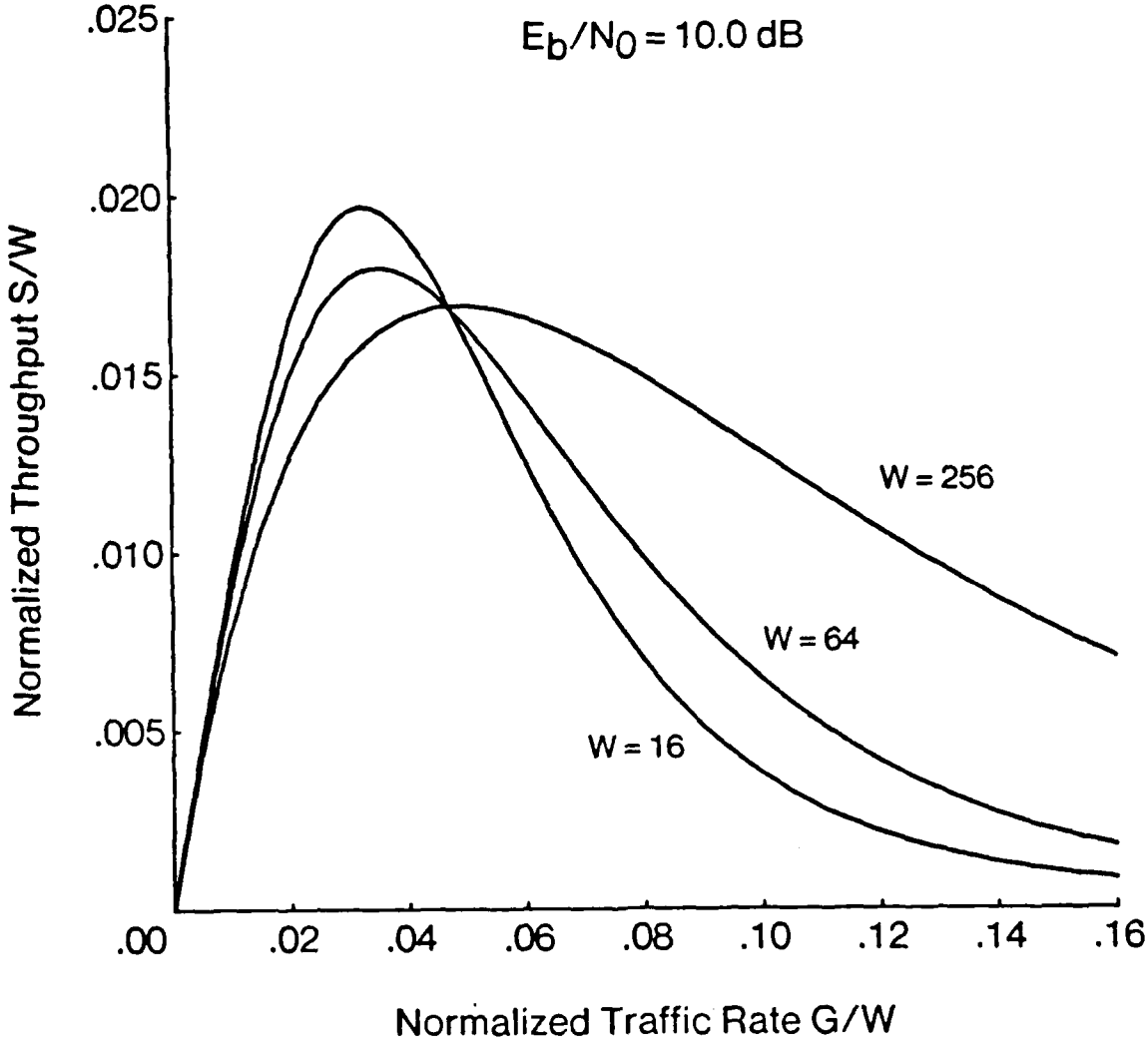


Fig. 4.4 Normalized throughput versus normalized traffic rate, infinite population. uncoded channel.

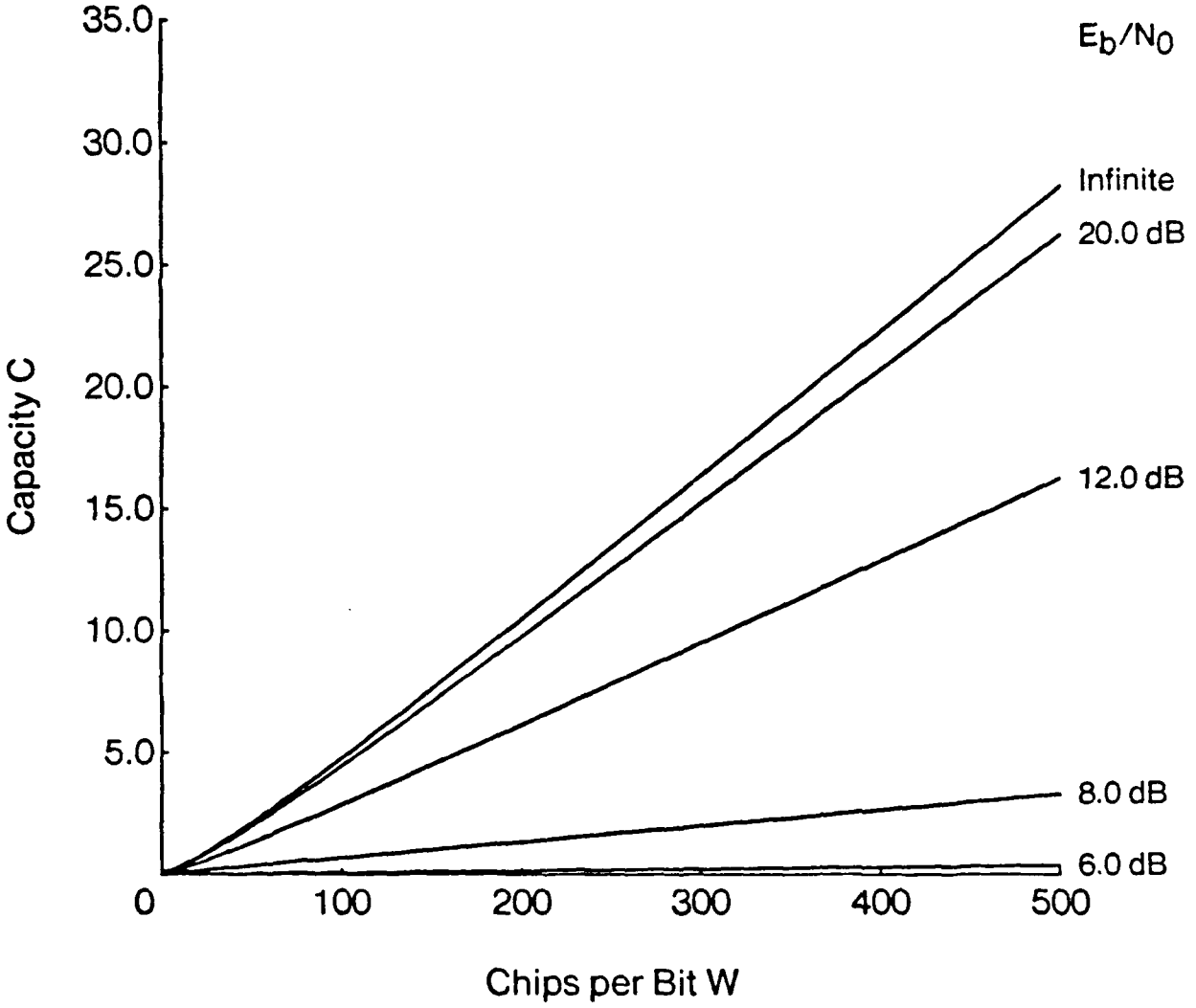


Fig. 4.5 Capacity versus spreading factor and received power, infinite population, uncoded channel.

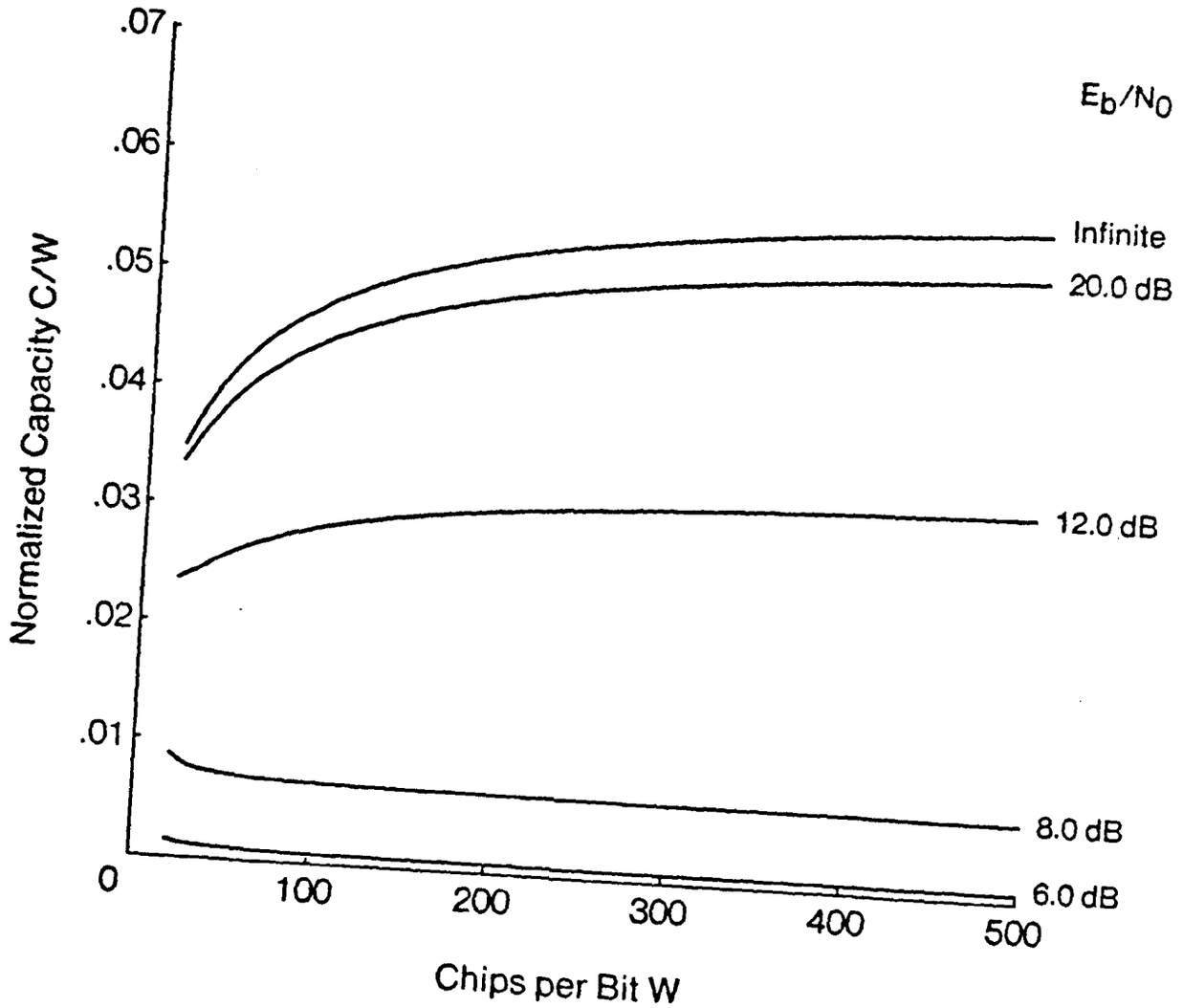


Fig. 4.6 Normalized capacity versus spreading factor and received power, infinite population, uncoded channel.

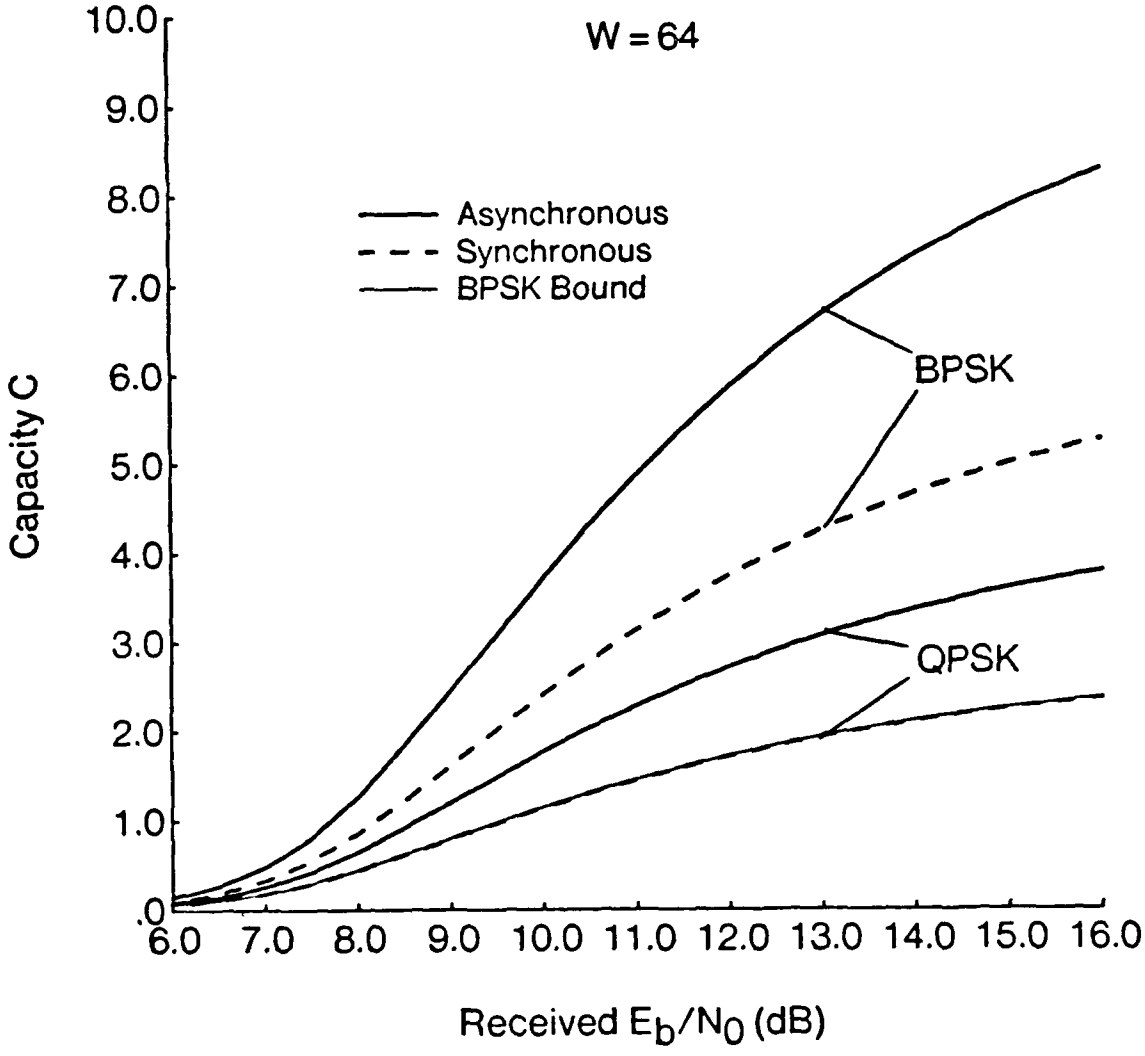


Fig. 4.7 Capacity versus received power for several types of coherent modulation, infinite population, uncoded channel.

Similar results are given for the non-coherent modulation formats in Fig. 4.8 for a spreading factor of $W = 64$. DPSK outperforms FSK, and the asynchronous channels outperform the synchronous ones. For an E_b/N_0 less than about 14.5 dB, synchronous DPSK does better than asynchronous FSK, but the ranking is reversed at higher E_b/N_0.

4.2 Finite Population Model

A more accurate model of the system is derived by considering a finite number of radios, each of which is either transmitting, receiving, or idle. Because all radios are identical, it is only necessary to keep track of the number of radios in a given state rather than the exact state of each individual radio. Radios are inhibited from transmitting when they are receiving a packet, so the state of the network at time t, $Z(t)$, must include the number of PRU's receiving, R, as well as the number transmitting, X. Therefore, Ω, the state space of $Z(t)$, is $\{(X,R) : 0 \leq X \leq M, 0 \leq R \leq X, \text{ and } X + R \leq M\}$, with a limiting state probability distribution $\{\pi_0(X,R), (X,R) \in \Omega\}$.

4.2.1 Primary Markov Chain

The state-transition rates for $Z(t)$ are determined in the following manner. For the state (X,R), new transmissions occur at an aggregate rate of $(M - X - R)\lambda$. This can be written as $M\lambda P_{SI}(X,R)$, where $P_{SI}(X,R)$ is the probability that a specific PRU is idle (i.e., not transmitting or receiving), given that the network state was (X,R) just prior to the start of a packet transmission. Because the radios are identical, for any user,

$$P_{SI}(X,R) = \frac{M - X - R}{M}. \tag{4.22}$$

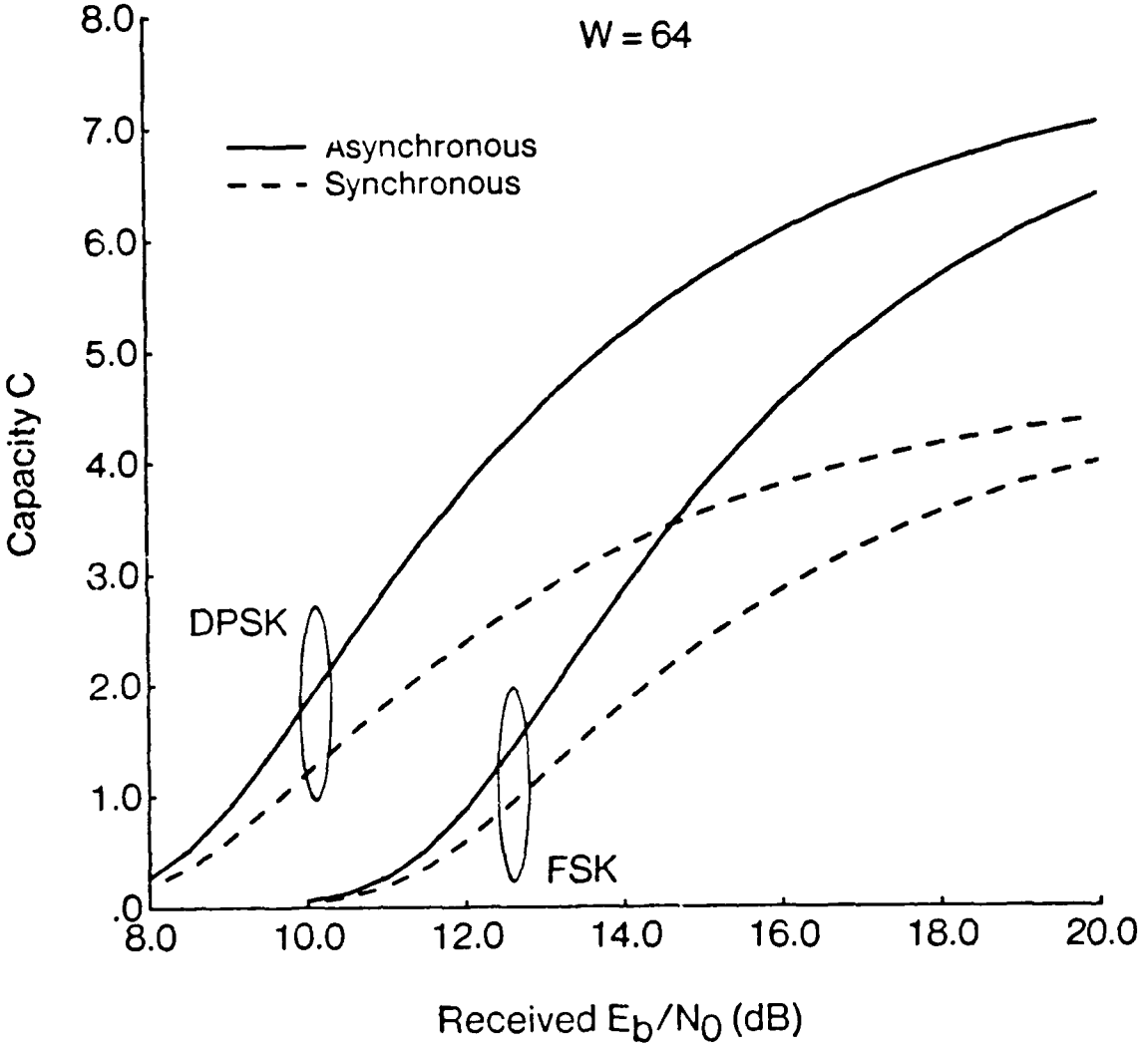


Fig. 4.8 Capacity versus received power for several types of non-coherent modulation, infinite population, uncoded channel.

Given that the source is idle and that a packet transmission begins, the packet will be received (possibly in error) if the destination is not transmitting or receiving and the destination receiver is able to acquire the packet (i.e., synchronize to the preamble). The probability that the packet is received conditioned on the source being idle and the packet being transmitted is denoted by $P_{Rx}(X,R)$. Corresponding to packets that are received, there are transitions to the state $(X+1, R+1)$ at rate $M\lambda P_{SI}(X,R)P_{Rx}(X,R)$. Corresponding to packets that are not received, there are transitions to state $(X+1, R)$ at rate $M\lambda P_{SI}(X,R)(1-P_{Rx}(X,R))$. Of the X packets being transmitted, R are being received and $X-R$ are not. Thus, due to the completion of packet transmissions, there is a transition to state $(X-1, R-1)$ at rate $R\mu$, and to state $(X-1, R)$ at rate $(X-R)\mu$. The state transitions for the general state (X,R) are diagrammed in Fig. 4.9, and the overall state space Ω is shown in Fig. 4.10.

At the network level, the model makes use of only the mean value of the probability of receiving a packet averaged over all possible network evolutions during the preamble, not the exact probability of receiving a specific packet with the particular network evolution encountered by that packet's preamble. This model relies on the assumption that the average probability of reception is sufficient to determine the behavior of a tagged packet. In reality, there will be correlation between the probabilities of receiving two packets whose preambles overlap in time. The model considers the average probability of receiving each independently. Because the performance measure of interest, the throughput, is a long term average, even though this correlation is not accounted for, the model should nevertheless yield a close approximation to the average throughput. Another implied approximation arises because the evolution of the network throughout the packet reception is considered to be independent from the event of the packet being received. In fact, there is some correlation between these events. However, if the preamble is short, the probability

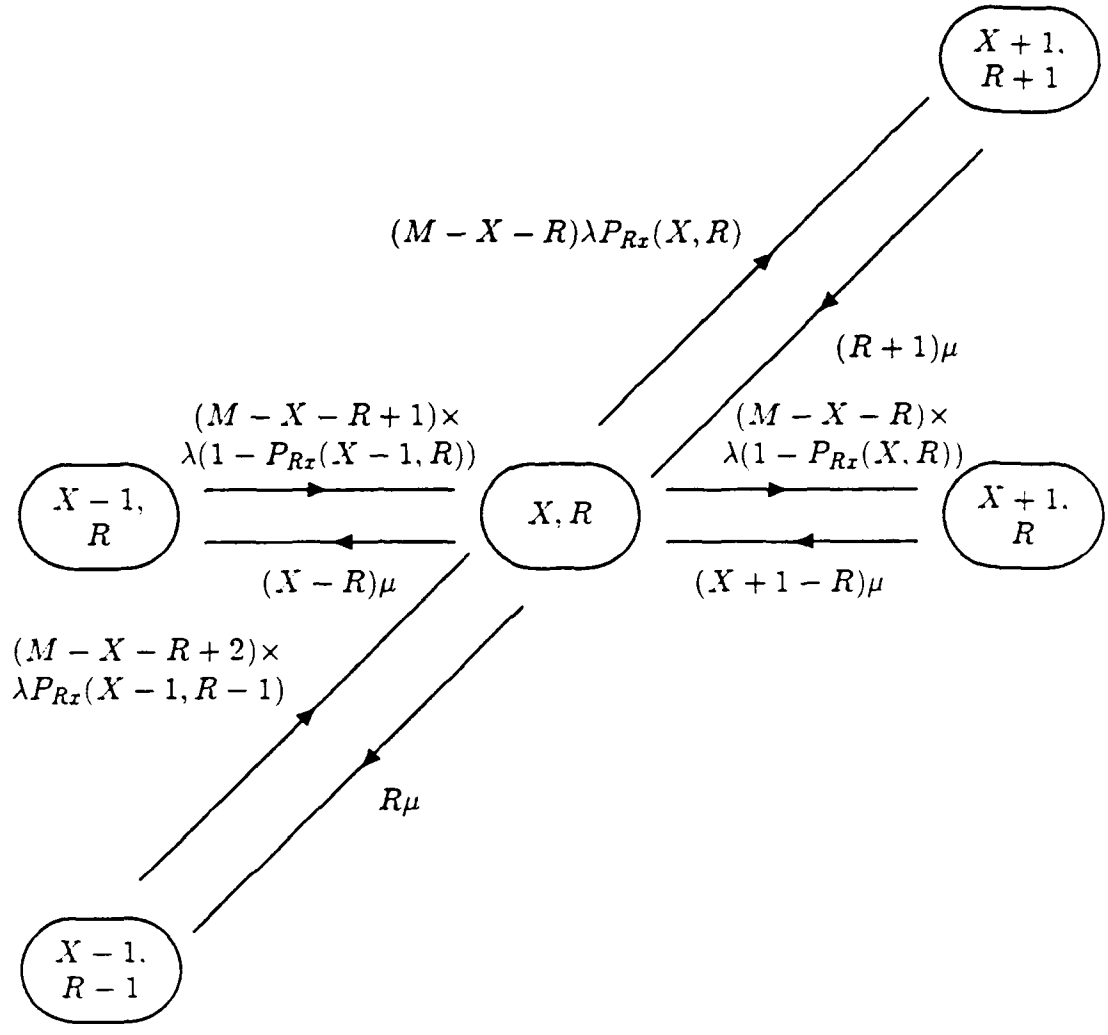


Fig. 4.9 Local state-transition-rate diagram for the finite population model.

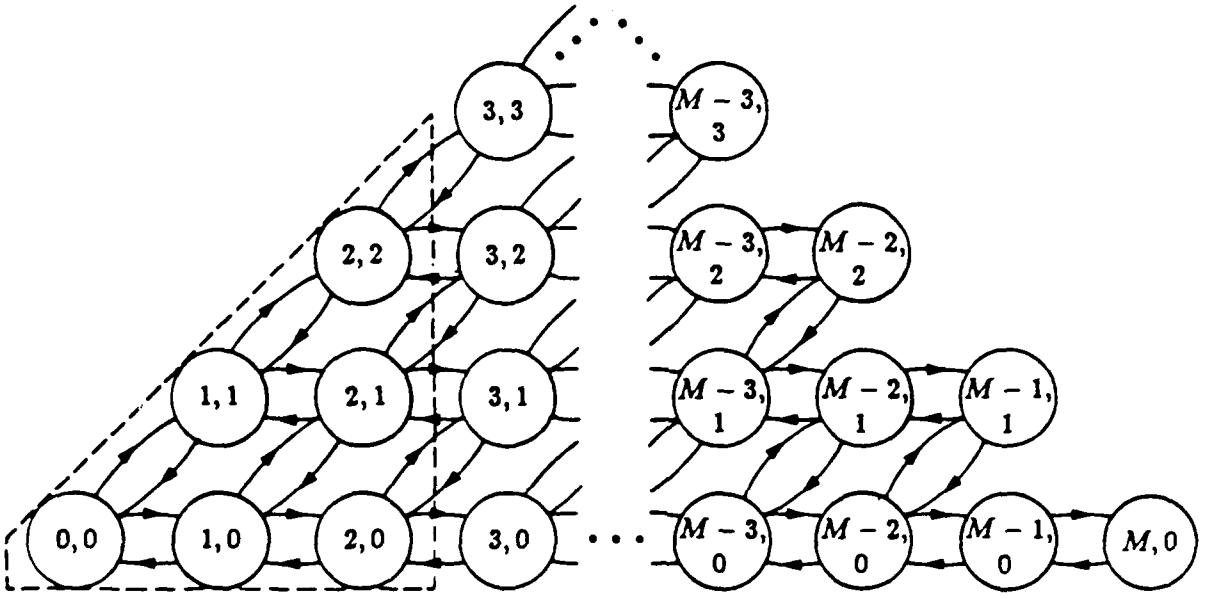


Fig. 4.10 State space for the finite population model.

of success is determined by the network states over a much longer interval than the time of the preamble, so the independence approximation will be good.

Therefore, even though the model may not exactly correspond to the network behavior, the error in network throughput resulting from the approximate model of preamble acquisition is expected to be very small. As will be discussed in detail in §6.4, the model has been validated by simulation, and the error was found to be less than about 6%. An accurate expression for $P_{Rx}(X,R)$ is derived in Chapter 6. Results given in this and the next chapter assume perfect synchronization. With this assumption, the probability that a packet is received is simply the probability that the destination is not transmitting or receiving at the start of the tagged packet, which is

$$P_{Rx}(X,R) = 1 - \frac{X+R}{M-1}, \tag{4.23}$$

since the destination is uniformly distributed over the $M-1$ other radios, $X+R$ of which are busy.

An analytical expression for $\pi_0(X,R)$, the steady state probabilities of the Markov chain, analogous to Eqn. 4.1 has not been found. Nevertheless, the probabilities can be found by solving the conservation equations,

$$\underline{\pi_0}\mathbf{Q} = \underline{0} \qquad \text{and} \qquad \sum_{(X,R)\in\Omega} \pi_0(X,R) = 1, \tag{4.24}$$

where $\mathbf{Q}$ again denotes the transition rate matrix.

4.2.2 Auxiliary Markov Chain

The state space Ω_{aux} of the auxiliary Markov chain $Z_{\text{aux}}(t)$ again consists of all the states that could be visited by the tagged packet, and two absorbing states,

Success and *Failure*. For this model, no cutoff L is necessary, as the state space is already finite. Therefore, Ω^*_{aux}, includes all those states of Ω for which $X \neq 0$. The states $(X, R) \in \Omega^*_{\text{aux}}$ are indexed by a linear ordering, $j = 1, 2, \ldots, J$, and the state *Success* is indexed as state $J+1$, and the state *Failure* as state $J+2$. This leads an expression identical to Eqn. 4.8,

$$\mathbf{Q}_{\text{aux}} = \begin{pmatrix} \mathbf{Q}^*_{\text{aux}} & \mu\underline{1}^T & \underline{\varphi}^T \\ \underline{0} & 0 & 0 \\ \underline{0} & 0 & 0 \end{pmatrix}, \tag{4.25}$$

where $\mathbf{Q}^*_{\text{aux}}$ is the $J \times J$ sub-matrix corresponding to transitions between the states $\{1, 2, ..., J\}$, whose elements can be derived from $\mathbf{Q}$ and the values of $\epsilon(X)$.

4.2.3 Throughput Calculation

The derivation of the throughput of the finite population model is similar to the presentation of §4.1.1. The throughput is defined in terms of the throughput per user, S_u, which is

$$S_u \triangleq \lim_{T \to \infty} \frac{1}{T}(\text{Time in interval } T \text{ spent by radio } u \text{ successfully transmitting data}). \tag{4.26}$$

$\overline{T}_s(X, R)$ is now defined as the expected contribution to throughput of a packet given that the network is in state (X, R) just prior to the start of transmission, the source is idle and begins transmitting, and the packet is received.

As mentioned previously, the correlation between the event of packet acquisition and the expected contribution to throughput is considered to be negligible. With this approximation, the throughput can be stated as the product of conditional probabilities,

$$S_u = \sum_{(X+1, R+1) \in \Omega^*_{\text{aux}}} \lambda \pi_0(X, R) P_{SI}(X, R) P_{Rx}(X, R)(\overline{T}_s(X, R) - P_{S|(X,R)} T_p). \tag{4.27}$$

where T_p is the length of the preamble. The term $P_{S|(X,R)}T_p$ is introduced to account for the overhead of the preamble. Because the packet distribution is exponential, there is a chance that it will be less that T_p, so the assumed overhead will be greater than the entire packet. However, if the preamble is less than 5% of the mean packet length, the probability of this event will be very small, so the inaccuracy will cause little error. The quantities $\overline{T}_s(X,R)$ and $P_{S|(X,R)}$ are found from Equations 4.15 and 4.16, where j corresponds to the state $(X+1, R+1)$. A rigorous derivation of the expression for raw throughput, which counts the overhead for the preamble as part of the throughput, is given in Appendix A.

Because all users are identical, the network throughput is M times S_u, which is

$$S = \sum_{(X,R)} \lambda\pi_0(X,R)(M-X-R)P_{Rx}(X,R)(\overline{T}_s(X,R) - P_{S|(X,R)}T_p). \qquad (4.28)$$

For perfect synchronization with $T_p = 0$, this simplifies to

$$S = \sum_{(X,R)} \lambda\pi_0(X,R)(M-X-R)\left(1 - \frac{X+R}{M-1}\right)\overline{T}_s(X,R). \qquad (4.29)$$

Even if both the channel and the scheduling of transmissions were perfect, the throughput would still be limited to $M/2$, since every successful packet requires both a source and a destination. Because the transmissions are random, some packets are wasted due to the destination being busy at the start of transmission. As the rate of traffic g increases, the number of wasted transmissions increases. Consequently, the capacity is achieved at lower values of g^*, which give an average number of transmissions that is smaller than $M/2$. When the spreading is large enough to give a small probability of bit error even for the worst interference that occurs with any significant probability at this rate of traffic, the performance will be very close to the performance with a perfect channel. Sousa and Silvester [47] have found an

analytical limit on throughput of $0.343(M/2)$ for M approaching infinity, assuming a perfect channel, perfect synchronization, and a negligible preamble size ($T_p = 0$). For the uncoded channel, the capacity is lower than this value. However, in the next chapter, it will be seen that for the coded channel, the probability of error can be negligible even for large M, so the user capacity of C/M approaches the limit of 0.172 for large M.

4.2.4 Results

The results given in this section are for a system with perfect synchronization, with $T_p = 0$. Fig. 4.11 is similar to Fig. 4.4 for the finite population with $M = 20$, except that the unnormalized rather than the normalized throughput is plotted. The plots are similar in shape, but the actual values are lower for the finite population. For a spreading factor of $W = 256$, the maximum throughput for the infinite population model is 5.02, while for the finite population model it is only 2.70. Because the receiver availability limits the throughput that can be achieved, the capacity levels off for high spreading factors. This is shown in Fig. 4.12, in which the capacity is plotted as a function of the spreading factor for several values of M, for the bound on the BPSK channel with $E_b/N_0 = 10.0$ dB. At small spreading factors, there is a linear increase of C with W, as was seen in the infinite population model, but at higher W the capacity levels off to a value that depends upon M. Consequently, there is a finite value of W which maximizes the normalized capacity C/W.

In Fig. 4.13, the network capacity C and user capacity C/M are plotted for even values of M as a function of the network size M for $W = 64$ and $W = 512$, with $E_b/N_0 = 10.0$ dB. For smaller W, the performance becomes limited by the channel at low values of M. The network capacity flattens out above $M \approx 10$, so the user

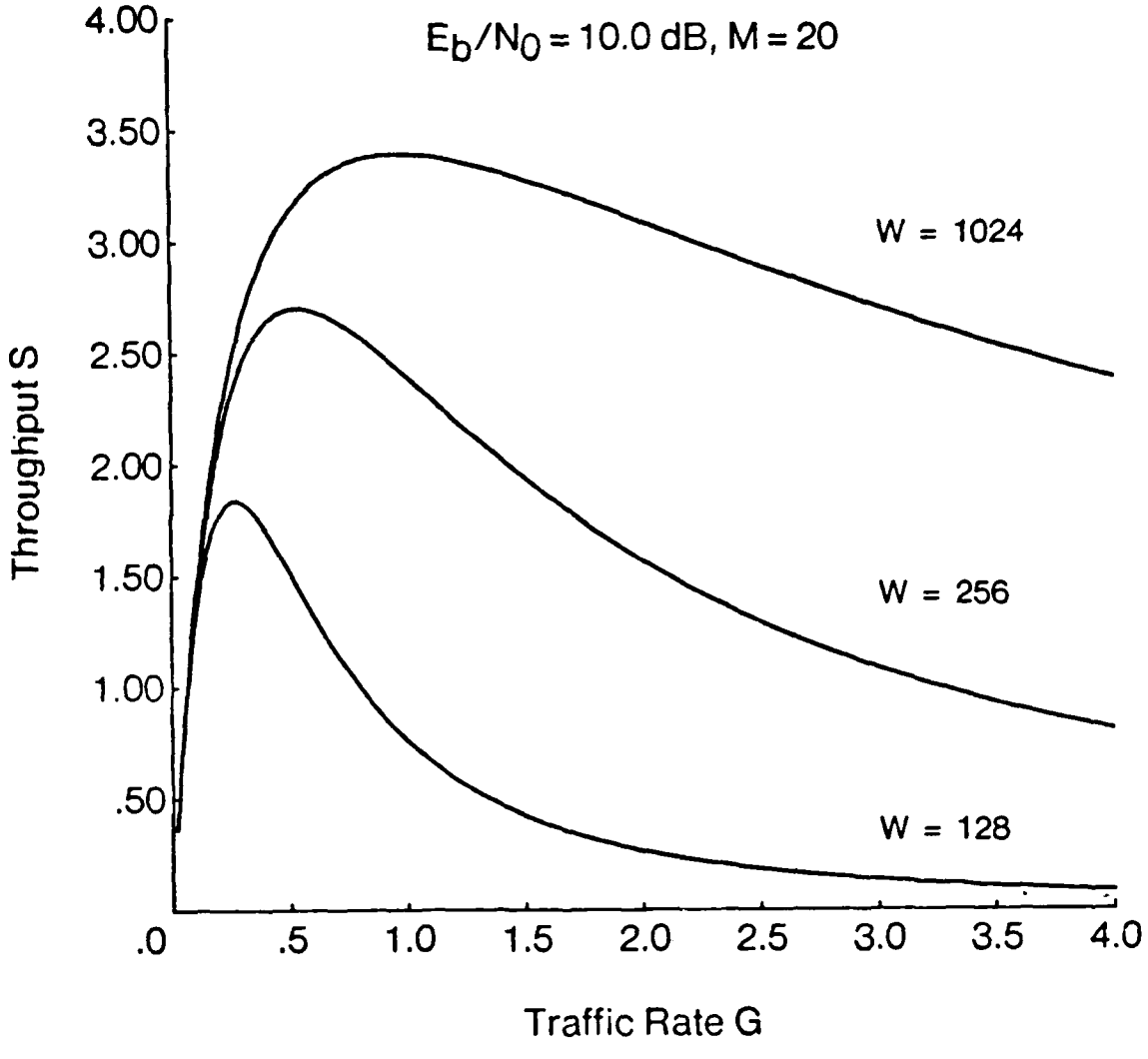


Fig. 4.11 Throughput versus traffic rate, finite population, uncoded channel.

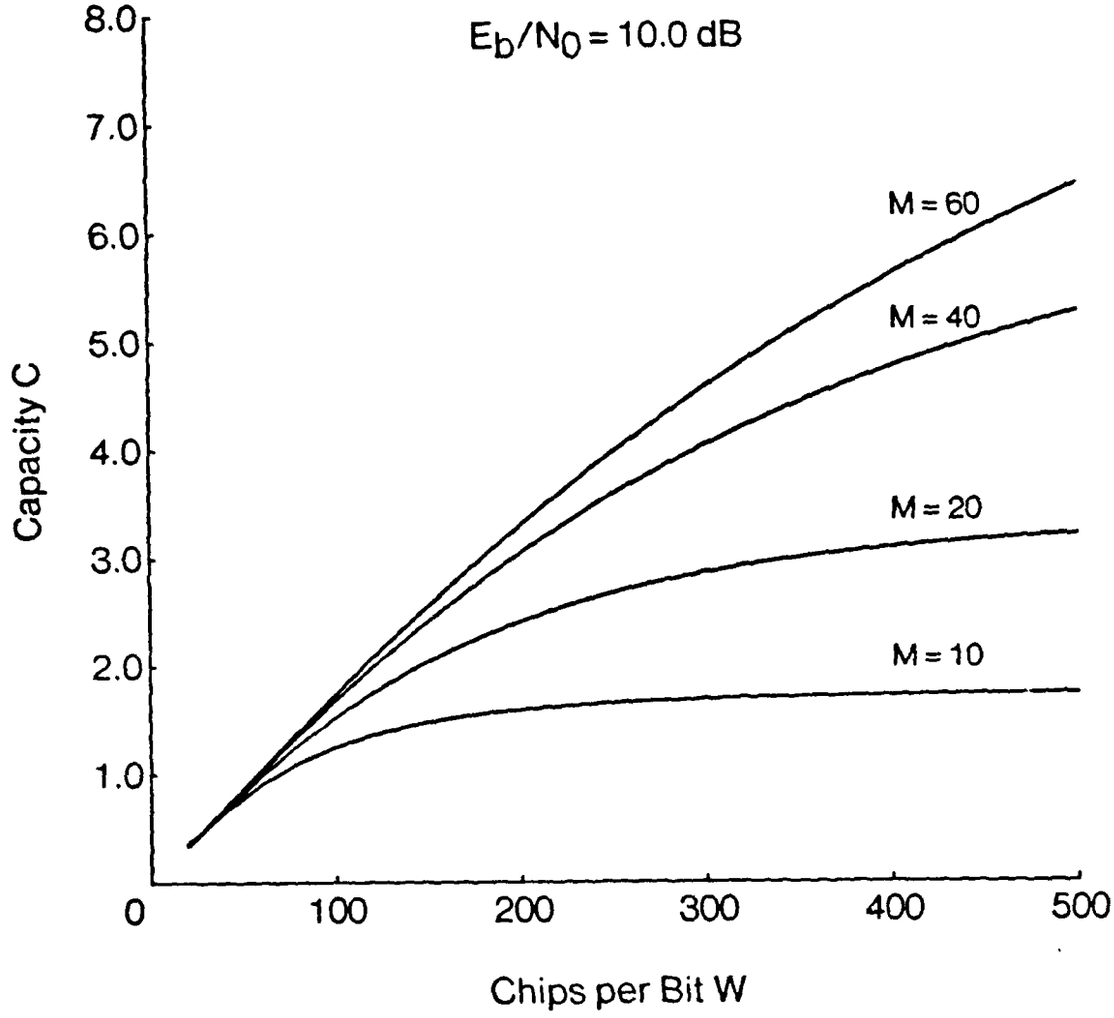


Fig. 4.12 Capacity versus spreading factor and network size, uncoded channel.

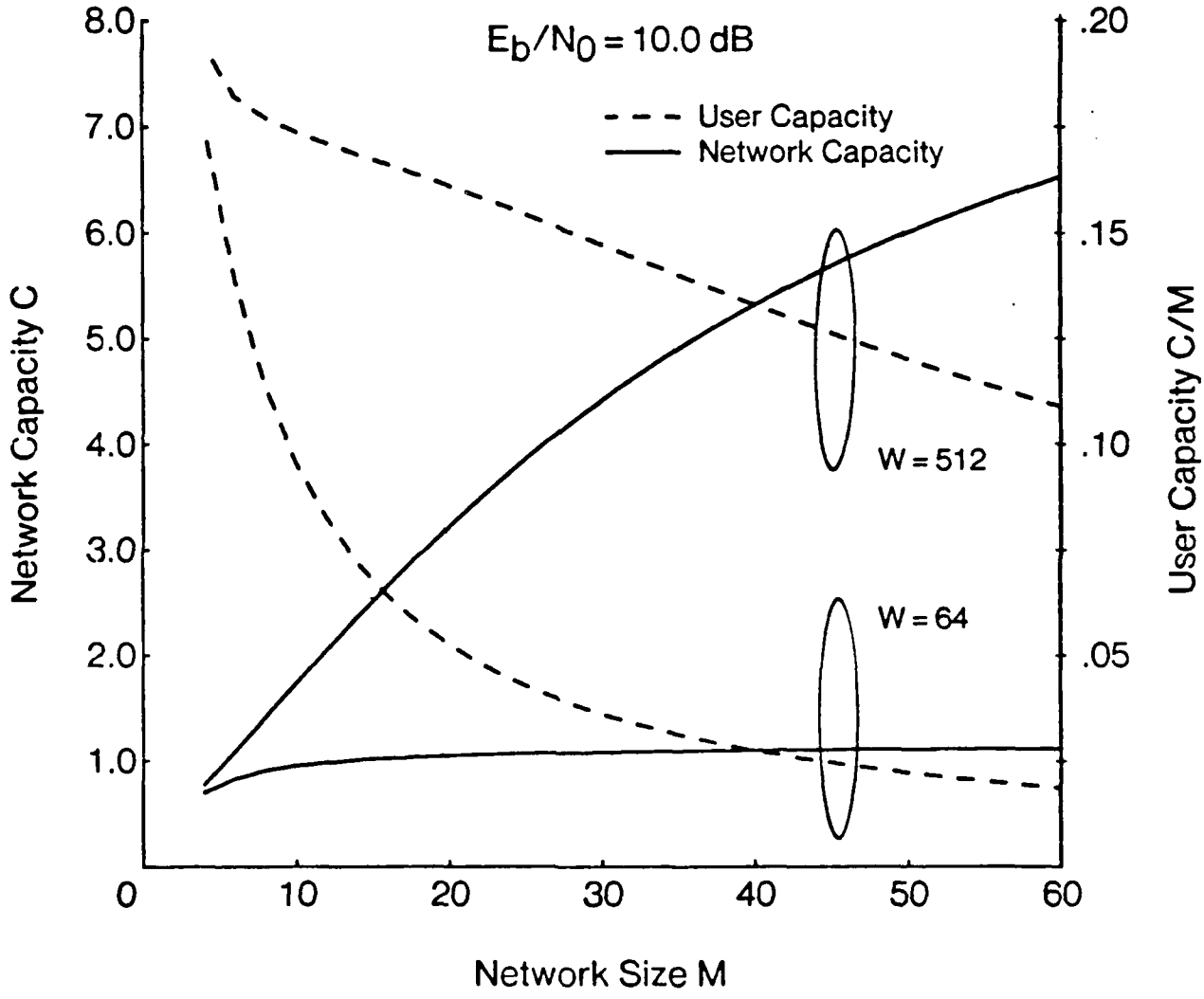


Fig. 4.13 Network and user capacity versus network size, uncoded channel.

capacity goes roughly as $1/M$. With a greater amount of spreading, the channel errors have a smaller effect, so the user capacity decreases slowly with increasing M.

4.3 Channel Load Sensing

Many authors have shown that the network performance of ALOHA systems can be improved by sensing the channel and blocking transmissions when the channel is busy. In an SSMA network where multiple users can transmit simultaneously, similar improvement may be possible if the radios can sense the number of radios transmitting and block transmission when the channel is heavily loaded. K is used to denote the number of sensed transmissions at which new transmissions are blocked.

It is assumed that each radio has perfect knowledge of the number of radios currently transmitting. In a real system, the radios can only estimate the channel loading. For example, if the code assignment for the preambles were space homogeneous, each radio could count the preambles detected in the recent past to estimate the number of other radios transmitting. Alternatively, the received signal can be integrated to give an estimate of the total energy in the signals received, which will indicate the channel loading. Clearly, these implementations will perform worse than the idealized model. Nevertheless, it is found that even with ideal sensing, except for small spreading factors, little improvement is gained by using channel load sensing. Therefore, the results indicate that the small improvement in the performance of a SSMA network from channel load sensing is not worth the cost and complexity of implementation.

In this section, the finite population model is modified to incorporate ideal channel sensing. Initially, it is assumed that the propagation delay between radios

is negligible, so that the knowledge of channel loading is acquired instantaneously. This assumption is later relaxed in an approximate model of non-zero propagation delay.

4.3.1 Zero Propagation Delay

With zero propagation delay, the modification of the model is straightforward. The change consists of deleting all transitions corresponding to new transmissions whenever the number of transmissions equals or exceeds K. However, this simple change can lead to a numerically unstable model, as discussed below. Therefore, a reformulation is necessary in order to find numerical results for a certain range of channel load sense point K and traffic rate g.

Because there will never be any more than M transmissions occurring, a channel load sense threshold $K \geq M$ will give the same performance as no channel load sensing. In the case of $K < M$ and $\epsilon(K)/\mu$ non-negligible, the state spaces of the Markov chains, Ω and Ω^*_{aux}, are truncated at $X = K$, but otherwise remain identical to the finite population model. For $K < M/2$ and $\epsilon(K)/\mu << 1$, almost all received packets will be successful. However, packets might not be received because of a busy destination or imperfect synchronization. Even so, with ideal synchronization, when K is small compared to M, $P_{Rx}(X, R)$ is always close to 1, so the throughput is maximized by a very large g^*. Unfortunately, the numerical solution used for finding the throughput becomes unstable as g becomes large. Therefore, an approximation of the Markov chain is used which is valid for such values of g.

For very large g, with high probability, the number of transmissions will be K, since as soon as a transmission ends a new scheduling point will be generated and transmission of a new packet will begin. However, in order to find the throughput,

the distribution of the number of captured receivers must be considered. In this section, it is proven that the probability of there being fewer than K active transmitters is $\mathcal{O}(1/g)$. Then, the distribution on the number of captured receivers is derived from an approximation of the Markov chain, which contains only the states $\{(X,R) : X = K-1 \text{ or } X = K\}$. $\overline{T}_s(K-1,R)$ is found from Eqn. 4.27 to give

$$S = \sum_{R=0}^{K} \frac{R\pi_0(K,R)}{(1+\epsilon(K)/\mu)^2}. \tag{4.30}$$

Consider the set of states (X,R) and $(X+1,R)$ for $X < M/2$. The marginal probability of there being X radios transmitting is denoted by $\hat{\pi}_0(X)$,

$$\hat{\pi}_0(X) = \sum_{R=0}^{X} \pi_0(X,R). \tag{4.31}$$

The expected number of users receiving given X is

$$E(R|X) = \sum_{R=0}^{X} R\frac{\pi_0(X,R)}{\hat{\pi}_0(X)}. \tag{4.32}$$

$\hat{\pi}_0(X)$ can be found from the balance equations resulting from a cut between states (X,R) and states $(X+1,R)$, shown in Fig. 4.10,

$$\begin{aligned} \sum_{R=0}^{X}(M-X-R)\lambda\pi_0(X,R) &= \sum_{R=0}^{X+1}(X+1)\mu\pi_0(X+1,R) \\ (M-X)\lambda\hat{\pi}_0(X) - \lambda\hat{\pi}_0(X)E(R|X) &= (X+1)\mu\hat{\pi}_0(X+1), \end{aligned} \tag{4.33}$$

so

$$\hat{\pi}_0(X) = \frac{(M-(X-1)-E(R|X-1))\lambda}{X\mu}\hat{\pi}_0(X-1). \tag{4.34}$$

$E(I|X)$ is defined to be the expected number of idle radios given X users transmitting,

$$\begin{aligned} E(I|X) &= E(M-X-R|X) \\ &= M-X-E(R|X), \end{aligned} \tag{4.35}$$

so

$$\hat{\pi}_0(X) = \frac{E(I|X-1)\lambda}{X\mu}\hat{\pi}_0(X-1)$$
$$= \left(\prod_{x=0}^{X-1}\frac{(\lambda/\mu)E(I|x)}{(x+1)}\right)\hat{\pi}_0(0). \quad (4.36)$$

For $X < M/2$, there will always be at least one idle radio, since $M - 2X \leq E(I|X) \leq M - X$. Therefore,

$$\frac{(\lambda/\mu)E(I|X)}{X+1} \quad \text{is } \mathcal{O}(g) \text{ for } X < M/2. \quad (4.37)$$

which implies that for $K \leq M/2$, $\hat{\pi}_0(K-1)/\hat{\pi}_0(K)$ is $\mathcal{O}\left(g^{-1}\right)$, and $\hat{\pi}_0(K-2)/\hat{\pi}_0(K)$ is $\mathcal{O}\left(g^{-2}\right)$. Thus, to first order, for large g, the approximation to the Markov chain which is shown in Figures 4.14 and 4.15 is valid.

Cut 1 indicated in Fig. 4.15 gives the local balance equation

$$(K-R)\mu\pi_0(K,R) = F(R)(1 - P_{Rx}(K-1,R));$$
$$F(R) = (M-K-R+1)\lambda\pi_0(K-1,R) \quad (4.38)$$

$F(R)$ is the rate of probability flowing out of state $(K-1,R)$, so cut 2 gives

$$F(R) = (K-R)\mu\pi_0(K,R) + (R+1)\mu\pi_0(K,R+1); \quad (4.39)$$

$$(R+1)\mu\pi_0(K,R+1) = \frac{(K-R)\mu\pi_0(K,R)}{1 - P_{Rx}(K-1,R)} - (K-R)\mu\pi_0(K,R); \quad (4.40)$$

$$\pi_0(K,R+1) = \frac{K-R}{R+1}\left(\frac{1}{1-P_{Rx}(K-1,R)} - 1\right)\pi_0(K,R)$$
$$\text{for } 0 \leq R \leq K-1. \quad (4.41)$$

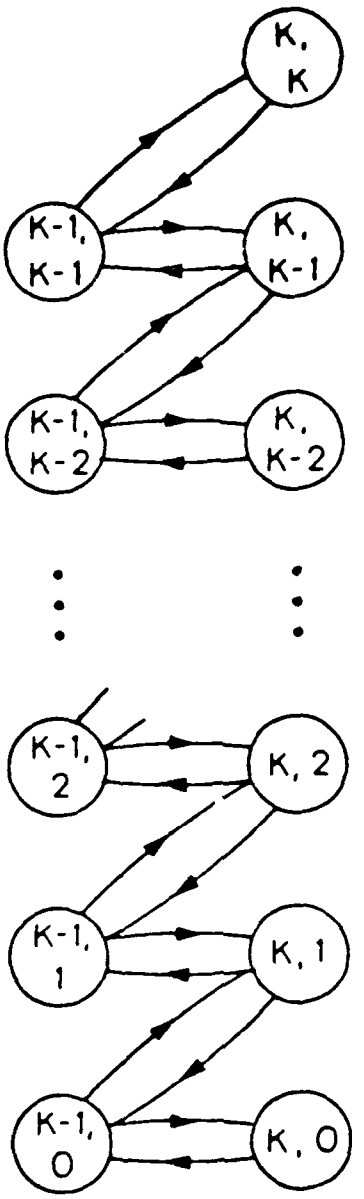


Fig. 4.14 State-transition diagram for the channel load sense model.

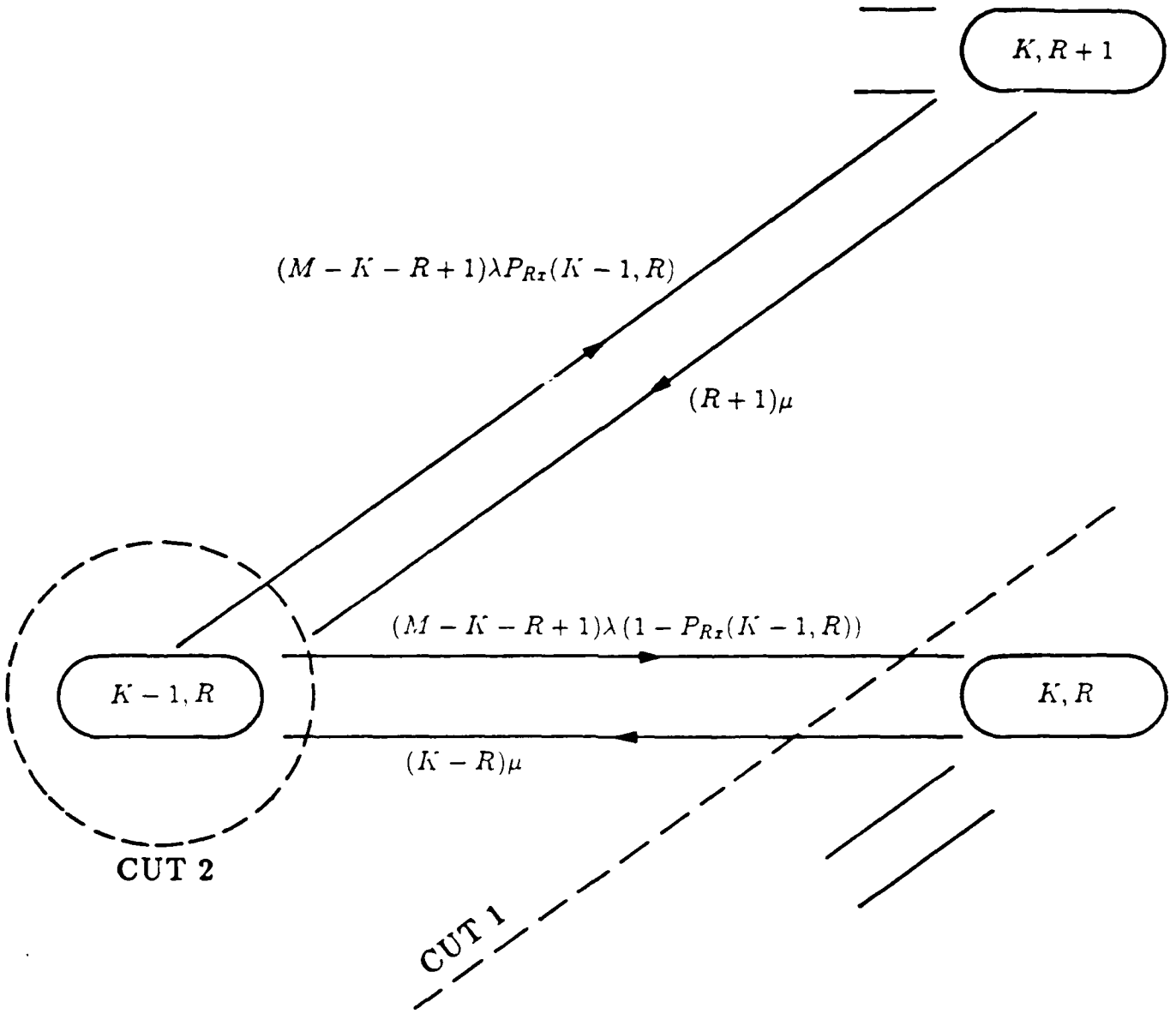


Fig. 4.15 Local state-transition-rate diagram for the channel load sense model.

From these recursion equations, the probabilities $\pi_0(K, R)$ can be found in terms of $\pi_0(K, 0)$, and then normalized by setting $\sum \pi_0(K, R) = 1$.

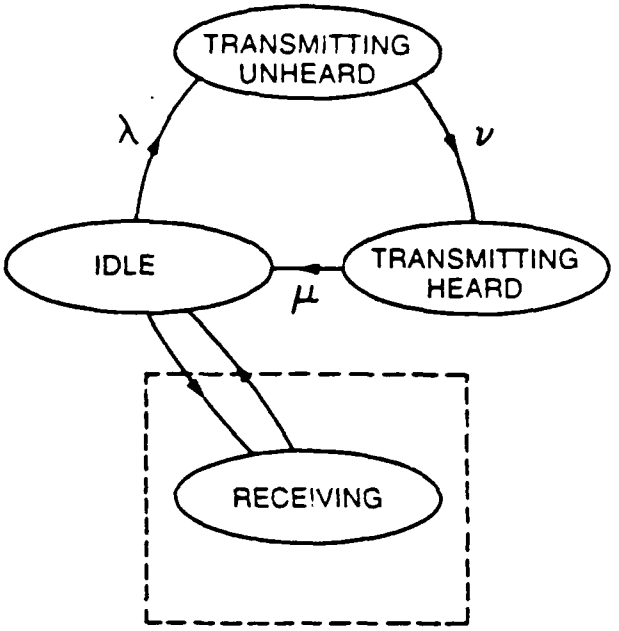
For the case when K is small compared to M, and $\epsilon_K/\mu << 1$, this approximation for the Markov chain gives the distribution $\pi_0(K, R)$, from which the throughput S can be derived. For larger K, the maximum throughput is achieved at values of g for which the approximation is no longer valid. However, for g in this range, the numerical solution described in §4.2 can be used.

4.3.2 Non-zero Propagation Delay

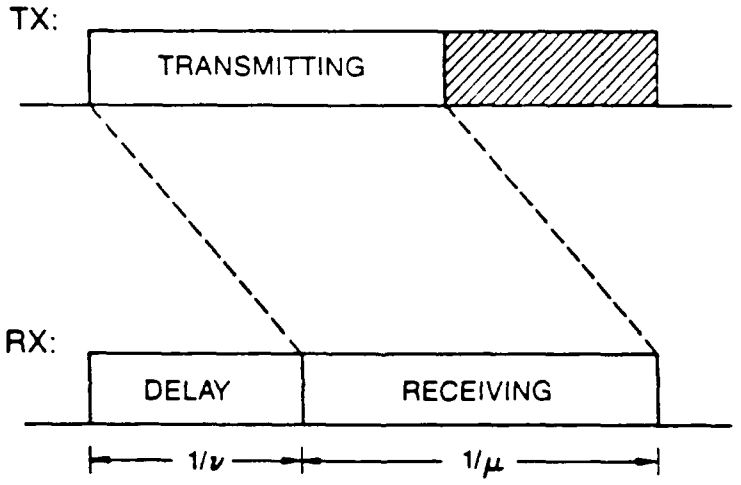
It is well known that carrier sensing systems degrade as the propagation delay becomes large relative to the packet lengths. An exact model of a fixed or varying propagation delay was found to be intractable. For this reason, the model used requires several unrealistic assumptions. However, even though the absolute results may not correspond to the performance of a real network, the model enables the relative performances of systems with and without channel load sensing to be compared as a function of the propagation delay.

The following assumptions lead to the approximate model of a system with non-zero propagation delay. A given packet requires an equal propagation time to reach all other users in the network. The propagation time is drawn from an exponential distribution, independently for each packet, regardless of the source or destination. In order that the propagation delay has the same effect at all users, including the source, the further pessimistic assumption is made that the transmitter is busy until the end of the packet has reached the destination.

A packet transmission is diagrammed in 4.16(a). The shaded region indicates this additional busy period for the transmitter. The propagation delay is modeled by introducing an exponentially distributed holding time, mean $1/\nu$, into every



(a)



(b)

Fig. 4.16 Non-zero propagation delay model.

transmission. The radios are modeled as in Fig. 4.16(b). A packet transmission now consists of two exponentially distributed periods. During the first period, the packet is not heard by any of the radios, so the state of the source radio is *transmitting, unheard.* During the second period, the packet is heard by all radios, and might be received, and the state of the source radio is *transmitting, heard.* The state space of the network model is therefore defined as (X_u, X_h, R), where X_u is the number of radios which are transmitting unheard, X_h is the number of radios which are transmitting heard, and R is the number of radios which are receiving. For the channel load sense models, the decision on whether or not to transmit is again based on the number of radios which can be sensed transmitting, X_h, but this decision does not take into account the radios that are transmitting but cannot be heard, X_u.

The state space of the primary Markov chain is the three-dimensional space indexed by X_u, X_h, and R;

$$\Omega = \left\{ \begin{array}{c} 0 \le X_u \le M, \\ (X_u, X_h, R) : 0 \le X_h \le M - X_u, \\ 0 \le R \le M - X_u - X_h, \text{ and } R \le X_h \end{array} \right\}.$$

Similarly, for the auxiliary Markov chain,

$$\Omega^*_{\text{aux}} = \left\{ \begin{array}{c} 0 \le X_u \le M, \\ (X_u, X_h, R) : 1 \le X_h \le M - X_u, \\ 1 \le R \le M - X_u - X_h, \text{ and } R \le X_h \end{array} \right\}.$$

The parameter h is defined to be ν/μ, so $1/h$ is the average holding time normalized by the mean packet length. $1/h$ is roughly equivalent to the standard delay measure a, which is the propagation delay normalized by the packet length.

For the state (X_u, X_h, R), with $X_h < K$, new transmissions at rate $(M - X_u - X_h - R)\lambda$ cause a transition to the state $(X_u + 1, X_h, R)$. For $X_h \ge K$, there are

no new transmissions. Radios change from transmitting, unheard, to transmitting, heard, at a rate ν, and may be received or not as in the finite population model, so there are transitions to the state $(X_u - 1, X_h + 1, R + 1)$ at rate $\nu P_{Rx}(X_u, X_h, R)$, and to state $(X_u - 1, X_h + 1, R)$ at rate $\nu(1 - P_{Rx}(X_u, X_h, R))$. With perfect synchronization, $P_{Rx}(X_u, X_h, R)$ is the probability that the destination is idle given state (X_u, X_h, R) and given that u is transmitting, unheard, which is

$$P_{Rx}(X_u, X_h, R) = \frac{M - X_u - X_h - R}{M - 1}. \tag{4.42}$$

Those packets which were received are completed at rate μR, causing transitions to $(X_u, X_h - 1, R - 1)$, and those that were not received finish at rate $\mu(X_h - R)$, with a transition to $(X_u, X_h - 1, R)$.

The user throughput S_u is

$$S_u = \sum_{(X_u - 1, X_h + 1, R+1) \in \Omega^*_{\text{aux}}} \nu \pi_0(X_u, X_h, R) P_H(X_u, X_h, R) P_{Rx}(X_u, X_h, R) \times (\overline{T}_s(X_u, X_h, R) - P_{S|(X_u, X_h, R)} T_p). \tag{4.43}$$

where $P_H(X_u, X_h, R)$ is the probability that user u is transmitting, unheard, given state (X_u, X_h, R), which is simply X_u/M. Thus, with ideal synchronization,

$$S = \sum_{(X_u - 1, X_h + 1, R+1) \in \Omega^*_{\text{aux}}} \nu \pi_0(X_u, X_h, R) X_u \times \left(\frac{M - X_u - X_h - R}{M - 1}\right) \overline{T}_s(X_u, X_h, R). \tag{4.44}$$

4.3.3 Channel Load Sense Results

The throughput S versus traffic rate g is plotted in Fig. 4.17 for a network size $M = 20$, using the bound on the BPSK $P_{e,symbol}$, $E_b/N_0 = 10.0$ dB, and $H = 64$.

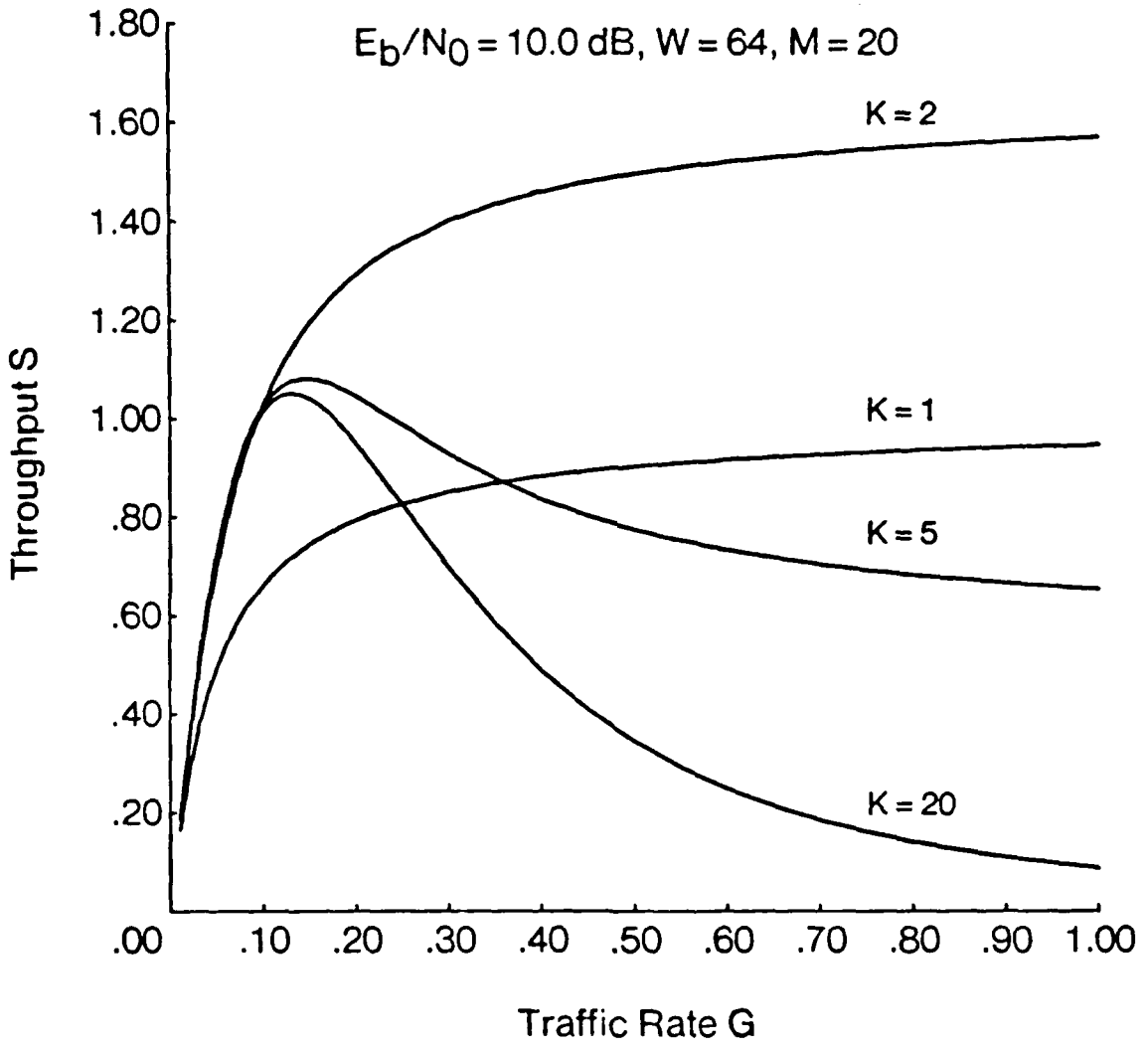


Fig. 4.17 Throughput versus traffic rate for channel load sensing, zero propagation delay, uncoded channel.

for the zero-propagation delay model with channel load sensing. For this channel, the capacity with $K = 2$ of 1.64 exceeds the capacity with no channel load sensing of 1.05 by 56%.

The capacity is plotted as a function of K for several values of W in Fig. 4.18 for the same network, with $E_b/N_0 = 10.0$ dB. For low values of W, the channel load sensing can give fairly substantial improvement in performance. The absolute increase is almost constant with respect to W, so for larger W there is a smaller percentage increase. Furthermore, this improvement decays for non-zero propagation delay, as shown in Fig. 4.19. The capacity C is plotted as a function of the delay parameter $1/h$ for $W = 64$ and $W = 512$, for no channel load sensing and for the channel load sense point K optimized at each value of h. For no channel load sensing, the throughput decreases as $1/h$ increases, due to the approximations of the model. The significant result is the relationship of throughput for load sensing to that with no load sensing. Even with ideal sensing, the improvement is small, especially as $1/h$ increases. In addition, this model does not account for hidden terminals, another factor that was found to severly degrade the performance of narrowband CSMA systems. Thus, the use of channel load sensing is expected to be much less beneficial for SSMA networks than for narrowband networks.

4.4 Summary

Several Markovian network models were developed in this chapter. The simplest model was the infinite population model, which did not account for receiver availability and modeled the aggregate traffic as Poisson. The state space of the more exact finite population model accounted for the states of the individual radios, and allowed the effect of receiver availability to be modeled. The model was modified to consider a system which uses a channel load sense protocol. The model was

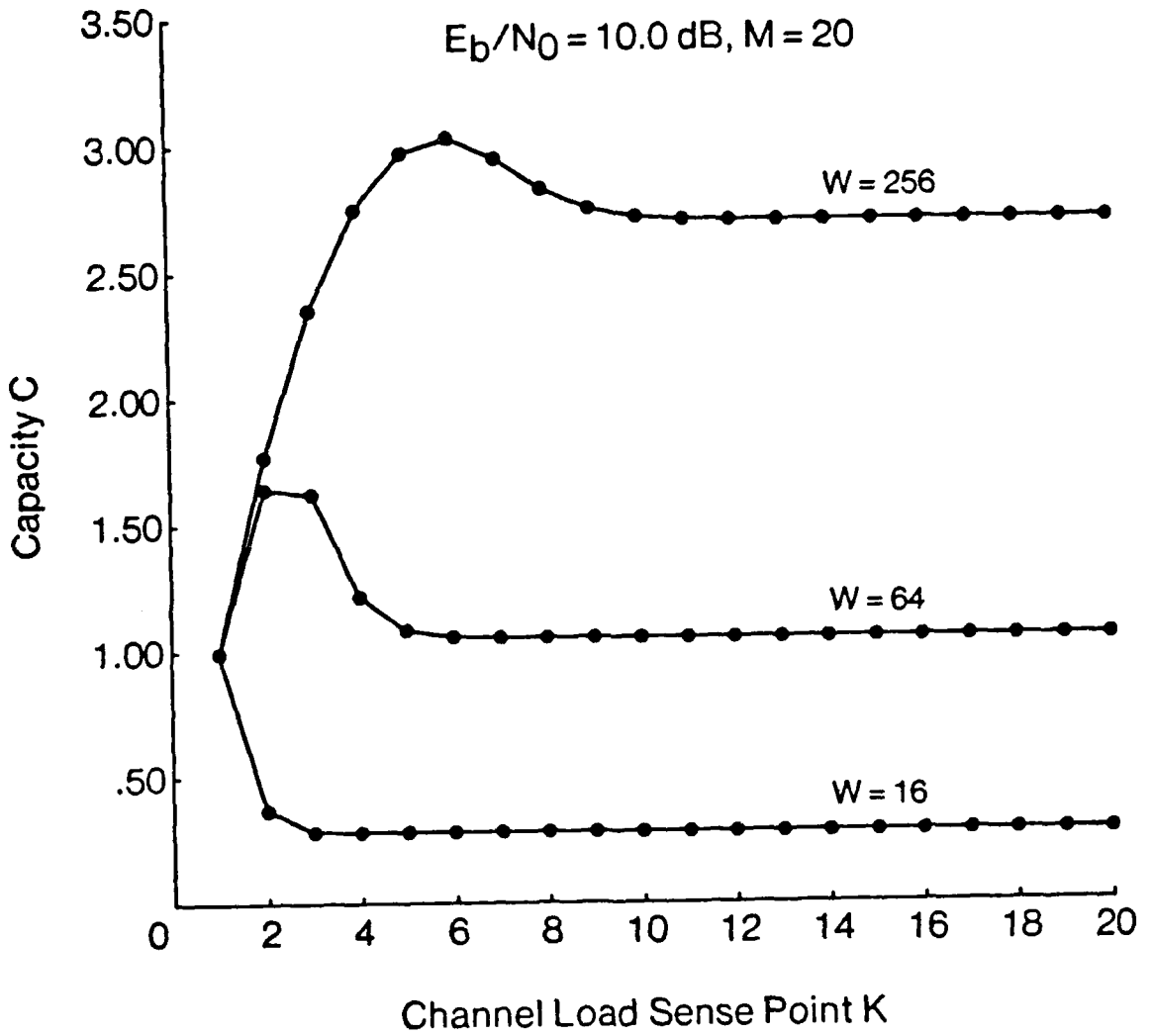


Fig. 4.18 Capacity versus channel load sense point, zero propagation delay, uncoded channel.

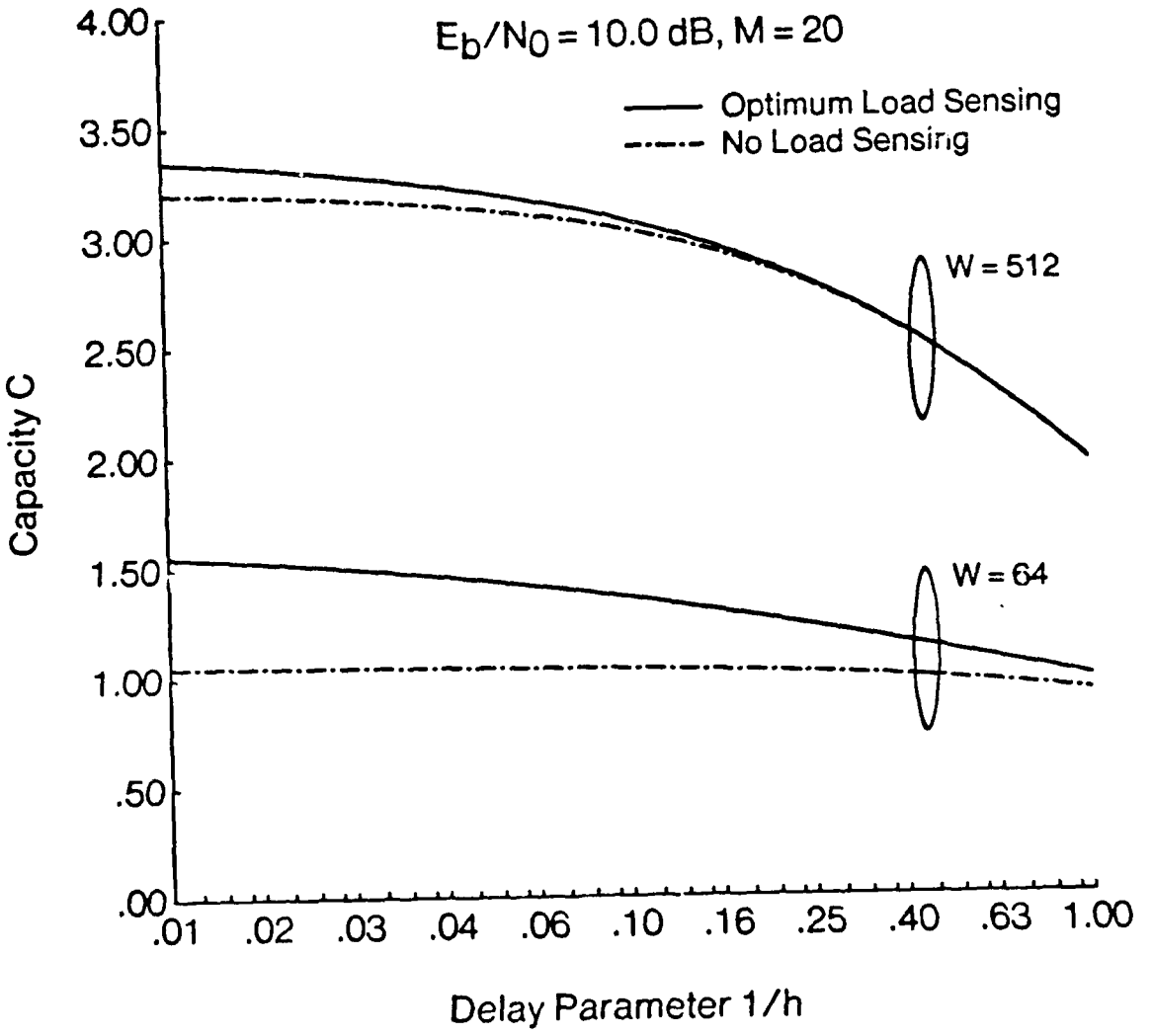


Fig. 4.19 Capacity versus propagation delay, channel load sensing, uncoded channel.

also extended to give an approximate model of a system with non-zero propagation delay. The general form for all of the models consisted of two Markov chains. The primary Markov chain gave the steady state probability of being in a given state, and the auxiliary Markov chain gave the probability of successful packet reception given the starting state. This approach allowed the channel models of Chapter 3 to be combined with the network model, leading to an integrated model of the SSMA PR network.

The throughput was found as a function of the traffic rate G or g, the received signal power E_b/N_0, the spreading factor W, and the network size M. For the channel load sense models, the additional parameters used were the sense threshold K and the delay parameter $1/h$, which is roughly equivalent to a, the ratio of maximum end-to-end propagation delay to mean packet transmission time. The numerical results indicate the effect of channel errors, evident at low spreading factors and low received signal powers. Several types of modulation were compared. Also shown was the effect of receiver availability. The use of channel load sensing led to improved performance for the ideal system, especially at low spreading factors. However, this improvement decreased in a system with appreciable propagation delay. Therefore, the results suggest that for an actual system, for which the sensing would not be ideal, the improvement will be minor, and will be unlikekly to justify the cost and complexity of channel load sensing hardware.

Chapter 5

Viterbi Decoder Analysis

The use of Forward Error Correction (FEC) coding will often lead to an improvement in the performance of SSMA PR networks. The network model of Chapter 4 can be used to study a system with FEC coding provided that the channel model can be formulated as a Poisson process. In this chapter, the performance of a Viterbi decoder in the environment of an unslotted SSMA PR network is analyzed. Although the Viterbi decoder has been extensively analyzed for a number of years, for two reasons these standard results are not applicable. First, they assume that the symbol error rate is constant, which is not the case for an unslotted SSMA network. Second, the aim of standard analyses is the long term average probability of bit error for a continuous stream of data, rather than the probability of success of a short packet. The analysis of this chapter yields the probability of success for a packet received in the presence of interference which varies over time. The resulting probability of success can be expressed in such a way as to give a Poisson process for the model of the generation of errors. The rates of this error process are evaluated numerically for a specific code. This allows the determination of network throughput, leading to results for a system with FEC coding which are similar to

those found in Chapter 4 for a system with an uncoded channel. The numerical results indicate explicitly the improvement in network throughput gained by FEC coding.

5.1 Introduction

In a system using FEC coding, even if the symbol errors are independent, the probability of bit errors in the output data stream will not be independent from bit to bit. Therefore, for an exact calculation of P_S, the probability that the packet is successful (i.e. that no errors occur in decoding), one must know the probability of error for each symbol of the packet. Theoretically, given the function $P_{e,symbol}(X)$ and the values of X throughout the entire packet, one could find P_S exactly. However, this would require a calculation of all possible codewords, and the probability of choosing each codeword given the values of $P_{e,symbol}(X)$ for each symbol, which quickly becomes infeasible for packets of more than a few bits. Instead, a bound on P_S is found which is a product over j_b of a bound on $P_C(j_b)$, the probability that no decoder error is made at a given bit position j_b. The result is similar to the bound derived by Taipale in [50], but a new derivation is found which is valid even when the symbol error rate varies during the reception of a packet.

From simulations of the decoder, it has been found that the decoder decision is essentially independent of symbol errors which occurred more than 4 or 5 constraint lengths K in the past [22], [30]. Therefore, the true bound on $P_C(j_b)$ is very likely to be bracketed by the results found by considering the maximum and minimum symbol error rates over the last $5K$ bits. This reduces the problem to one of calculating $P_C(j_b)$ for a constant symbol error rate. For this problem, the bound on $P_C(j_b)$ can be expressed as a finite product. In §5.4.1, the terms of this product

are given for the specific case of the rate 1/2, constraint length $K = 7$ code with generator polynomial (in Octal) 171, 133 [34].

For a system in which the probability of symbol error changes slowly over $5K$ bits, the bound on $P_C(j_b)$ can be approximated by finding the value of $P_C(j_b)$ resulting from a constant probability of symbol error equal to the current value. The resulting model is both memoryless and independent from bit to bit, so the Poisson error model assumed in Chapter 3 can be used even for the coded channel.

5.2 Analysis

Unlike block codes, convolutional codes do not require a frame structure. The output symbols from the encoder at any time are a linear combination (modulo 2) of the previous K bits, so the output stream depends on the input bits in a sliding window fashion. For a rate m/n code, n symbols are output for each m bits. Typically, for $m = 1$ codes, a shift register stores the last K bits, and each of the n symbols is generated by a different linear binary function of these bits. The major parameters of the code are its constraint length K and its rate m/n. In this dissertation, numerical results are found only for the rate 1/2, constraint length 7 code, and for illustrative purposes a rate 1/2, constraint length 3 code is also discussed. The approach taken can be used to study other convolutional codes, with different rates and constraint lengths.

The following terminology is used. Bits refer to the uncoded data stream input from the end user. Symbols refer to the binary sequence output from the encoder (for a rate 1/2 code, there are two symbols for every bit). The word position refers to the bit or symbol position in the data sequence, and the word bit or symbol is

generally reserved to mean the value (0 or 1) occurring at a given position. The bit position is denoted by j_b, and the symbol position by j_s.

The analysis considers a tagged packet of length $\mathcal{L}+K-1$ bits, which consists of $\mathcal{L}$ data bits and a known trailer of $K-1$ bits. The encoder is preset to a known state at the start of each packet. The results of this chapter apply to any channel that can be characterized by a sequence of states p_{j_s}, corresponding to the probabilities of symbol error at each symbol position j_s, and for which the symbol errors are independent given the sequence of states throughout the packet. This sequence of states, $(p_1, p_2, \ldots, p_{2(\mathcal{L}+K-1)})$, will be denoted by $\mathcal{P}$. For the system under consideration, the probability of symbol error is derived in the same manner as the probability of bit error in §3.4, with the word "symbol" replacing the word "bit" everywhere. The model of symbol errors given in §3.4 leads to a memoryless binary symmetric channel, for which the probability of symbol error is a function of the current channel state, so $p_{j_s} = P_{e,symbol}(X_{j_s})$, where X_{j_s} is the network state X at the j_sth symbol. Also, for this model of the SSMA radio channel, the symbol errors are conditionally independent given $\mathcal{P}$, so the requirements stated earlier are satisfied. Throughout this chapter all probabilities are conditioned on $\mathcal{P}$. For notational simplicity, this condition is not explicitly stated. Because the channel errors are symmetric, and convolutional codes are group codes, P_S is the same for any transmitted codeword [34], so the data bits, trailer, and starting state are chosen to be all zeros.

$\mathcal{D}_\alpha$ is a sequence of $\mathcal{L}+K-1$ bits, where the first $\mathcal{L}$ bits are arbitrary and the last $K-1$ bits are the known trailer sequence, and $\alpha \in \{0, \ldots, 2^{\mathcal{L}}-1\}$. $\mathcal{D}_\alpha^{(j_b)}$ is the subsequence consisting of the first j_b bits of the sequence $\mathcal{D}_\alpha$. $D_\alpha(j_b)$ is the j_bth bit of $\mathcal{D}_\alpha$. $\mathcal{C}_\alpha$ is the sequence of $2(\mathcal{L}+K-1)$ symbols output from the encoder (with a rate 1/2 code), for an encoder input of $\mathcal{D}_\alpha$. Thus, $\{\mathcal{C}_\alpha, \alpha = 0, \ldots, 2^{\mathcal{L}}-1\}$ is the set of all

possible codewords for a packet with $\mathcal{L}$ data bits. $\mathcal{C}_\alpha^{(j_s)}$ is the subsequence consisting of the first j_s symbols of the codeword $\mathcal{C}_\alpha$ ($j_s \equiv 2j_b$). $\mathcal{C}_\alpha(j_s)$ is the j_sth symbol of $\mathcal{C}_\alpha$. The all zero data sequence is $\mathcal{D}_0$, and the corresponding codeword, which is the transmitted codeword, is $\mathcal{C}_0$. Given two codewords, $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$, $|\mathcal{C}_\alpha - \mathcal{C}_\beta|$ is the Hamming distance between $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$. $\mathcal{S}_R$ is the sequence of $2(\mathcal{L}+K-1)$ received symbols, which is equal to the sequence of channel errors. $\mathcal{S}_R^{(j_s)}$ is the subsequence consisting of the first j_s received symbols, and $S_R(j_s)$ is the j_sth symbol of $\mathcal{S}_R$. $\delta_H(\mathcal{C}_\alpha^{(j_s)})$ is the difference in Hamming distances, $|\mathcal{S}_R^{(j_s)} - \mathcal{C}_\alpha^{(j_s)}| - |\mathcal{S}_R^{(j_s)} - \mathcal{C}_0^{(j_s)}|$. $\mathcal{E}$ denotes an arbitrary encoder state. $\mathcal{S}_\mathcal{E}^{(j_b)}$ is the survivor path for state $\mathcal{E}$ at position j_b.

The encoding function is the mapping of the $\mathcal{L}+K-1$ bit sequence $\mathcal{D}_\alpha$ to a $2(\mathcal{L}+K-1)$ symbol codeword $\mathcal{C}_\alpha$. The encoder can be modeled as a finite state machine. Transitions between states occur at each new input bit. For a rate 1/2, constraint length K convolutional code, the symbols at positions $j_s = 2j_b - 1$ and $j_s = 2j_b$ only depend upon the bits $D_\alpha(j_b - K + 1), D_\alpha(j_b - K + 2), \ldots, D_\alpha(j_b)$ of the data sequence. Because the oldest bit of the K bits in the shift register does not affect the next state, only 2^{K-1} states are needed. Thus, at position j_b, the state of the encoder $\mathcal{E}$ is defined to be the $K-1$ bits $D_\alpha(j_b-K+2), D_\alpha(j_b-K+3), \ldots, D_\alpha(j_b)$. The state $\mathcal{E}$ and the value of the bit at position j_b+1 uniquely determine both the output and the next state. The finite state machine representation of the rate 1/2, $K = 3$ encoder is shown in Fig. 5.1, which also indicates the two symbols output at each transition.

For the sake of brevity, the phrase "$\mathcal{C}_\alpha$ is in state $\mathcal{E}$ at position j_b" is used to mean that $\mathcal{E}$ is the state at position j_b of the encoder which produced $\mathcal{C}_\alpha$. The encoder is said to be in the zero state at bit j_b when $D_\alpha(j_b - K + 2) = 0, D_\alpha(j_b - K + 3) = 0, \ldots,$ and $D_\alpha(j_b) = 0$. Also, the encoder is said to return to the zero state at bit

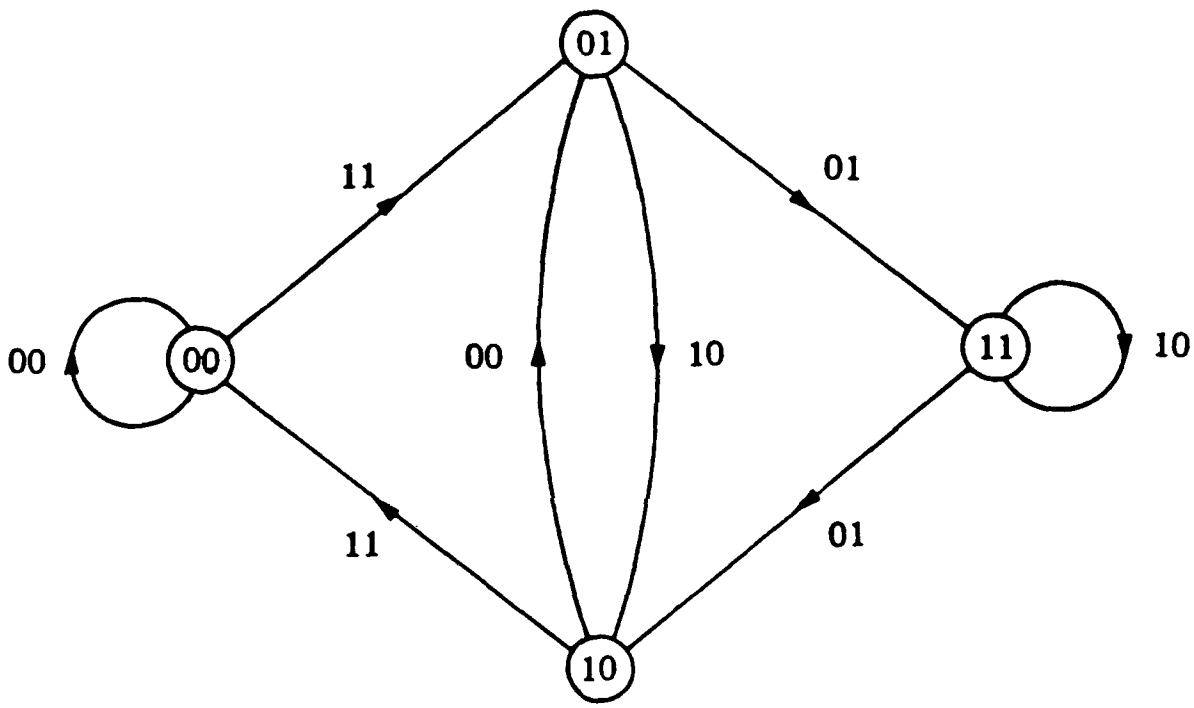


Fig. 5.1 Finite state machine model. Rate 1/2, K=3 code.

j_b when the encoder is not in the zero state at bit $j_b - 1$ and is in the zero state at bit j_b.

The Viterbi decoding algorithm is summarized below. A much more detailed treatment can be found in the paper by Viterbi [51] and also in most texts on error correction coding, such as Clark and Cain [10] and Blahut [3]. At each bit position j_b, which corresponds to symbol position $j_s = 2j_b$, for each of the 2^{K-1} states $\mathcal{E}$ of the encoder, the decoder stores a sequence of encoder states which begins in the known starting state and ends in the state $\mathcal{E}$. Each such sequence of states is referred to as a survivor path, denoted by $S_{\mathcal{E}}^{(j_b)}$. Each survivor path can be associated with a data sequence $\mathcal{D}_\alpha^{(j_b)}$, so the notation $S_{\mathcal{E}}^{(j_b)} = \mathcal{D}_\alpha^{(j_b)}$ is used to mean that the path $S_{\mathcal{E}}^{(j_b)}$ results when the first j_b bits of the encoder input are $\mathcal{D}_\alpha^{(j_b)}$. If $\mathcal{C}_\alpha^{(j_s)}$ is the codeword that would be output for a sequence of encoder states corresponding to $S_{\mathcal{E}}^{(j_b)}$ (i.e. $S_{\mathcal{E}}^{(j_b)} = \mathcal{D}_\alpha^{(j_b)}$), then the Hamming distance between $\mathcal{S}_R^{(j_s)}$ and $\mathcal{C}_\alpha^{(j_s)}$ is less than (or equal to) the Hamming distance between $\mathcal{S}_R^{(j_s)}$ and any other codeword which begins in the starting state and ends in state $\mathcal{E}$ at position j_b. When several codewords are at equal Hamming distance from $\mathcal{S}_R^{(j_s)}$, one of the associated paths is chosen randomly.

In addition to the survivor paths, the decoder stores a metric for each of the 2^{K-1} codewords associated with the survivor paths. For the hard decision decoder being considered, the metric is the Hamming distance $|\mathcal{S}_R^{(j_s)} - \mathcal{C}_\alpha^{(j_s)}|$. Consider the two extensions of the survivor path $S_{\mathcal{E}}^{(j_b)} = \mathcal{D}_\alpha^{(j_b)}$ resulting from appending first a 0 and then a 1 to $\mathcal{D}_\alpha^{(j_b)}$. There are now twice as many paths to consider. However, each of the new paths can be paired with another which ends in the same state at bit $j_b + 1$. Let two such paths be $\mathcal{D}_\beta^{(j_b+1)}$ and $\mathcal{D}_\gamma^{(j_b+1)}$, which both end in state $\mathcal{E}$ at bit $j_b + 1$. From the information stored regarding the survivor paths at bit j_b, the metric of each path is known up to bit j_b. The values of the

two symbols $S_R(2j_b - 1)$ and $S_R(2j_b)$, are also known, and the codeword symbols $C_\beta(2j_b - 1)$, $C_\beta(2j_b)$, $C_\gamma(2j_b - 1)$, and $C_\gamma(2j_b)$, can be calculated, so the metrics of paths $\mathcal{D}_\alpha^{(j_b+1)}$ and $\mathcal{D}_\beta^{(j_b+1)}$ can be computed. The path with the better metric is chosen as the new survivor path $S_\mathcal{E}^{(j_b+1)}$. Also, the metrics of the new survivor paths are stored. It can be seen that this algorithm will never eliminate the codeword which is at minimum Hamming distance from $\mathcal{S}_R$, so the decoding algorithm yields the maximum likelihood codeword.

Because a known trailer is transmitted, at the end of the packet the decoder can choose one unique codeword from the 2^{K-1} survivor paths, namely, the one which ends in the state specified by the trailer. The packet is successfully decoded if and only if $S_0^{(\mathcal{L}+K-1)}$, the survivor path for the zero state, is equal to the all zero path $\mathcal{D}_0$, so

$$P_S = \prod_{j_b=1}^{\mathcal{L}+K-1} \Pr(S_0^{(j_b)} = \mathcal{D}_0^{(j_b)} | S_0^{(j_b-1)} = \mathcal{D}_0^{(j_b-1)}). \tag{5.1}$$

$P_C(j_b|1, 2, \ldots, j_b - 1)$ is defined to be the conditional probability*

$$P_C(j_b|1, 2, \ldots, j_b - 1) = \Pr(S_0^{(j_b)} = \mathcal{D}_0^{(j_b)} | S_0^{(j_b-1)} = \mathcal{D}_0^{(j_b-1)}). \tag{5.2}$$

To find $P_C(j_b|1, 2, \ldots, j_b - 1)$, it is only necessary to consider those codewords which return to the zero state at position j_b. Furthermore, the condition implies that any of these codewords which have previously returned to the zero state have already been eliminated, so the only codewords that could be chosen as $S_0(j_b)$ are those which return to the zero state for the first time at position j_b. A partial codeword $\mathcal{C}_\alpha^{(2j_b)}$ which eliminates $\mathcal{C}_0$ at position j_b can have many future evolutions, that is, many codewords $\mathcal{C}_\alpha$ can begin with the sequence $\mathcal{C}_\alpha^{(2j_b)}$. However, the evolution of the partial codeword after it eliminates $\mathcal{C}_0$ is irrelevant to a determination of the

*P_C is $1 - P_E$, where P_E is the probability of first-event error as in [51]

success of a packet. Consequently, only the unique codeword $\mathcal{C}_\alpha$ which equals $\mathcal{C}_\alpha^{(j_s)}$ for symbols $j_s \leq 2j_b$ and is zero for symbols $2j_b < j_s \leq 2(\mathcal{L}+K-1)$ is considered. The set of codewords which return to the zero state at position j_b and remain in the zero state thereafter is denoted by $\psi(j_b)$. Even for the data sequence of a single one followed by all zeros, the zero state is not reached until position K, so $\psi(j_b) = 0$ for $j_b < K$. Ψ will denote the union over $j_b = K, K+1, \ldots, \mathcal{L}+K-1$ of $\psi(j_b)$. Also, any codeword $\mathcal{C}_\alpha \in \psi(j_b)$, is identically zero for positions after j_b, so $\delta_H(\mathcal{C}_\alpha^{(j_b)}) = \delta_H(\mathcal{C}_\alpha)$.

If $\delta_H(\mathcal{C}_\alpha) > 0$ for every $\mathcal{C}_\alpha \in \Psi$ then $\mathcal{D}_0$ will be chosen as the survivor path at every position, so the packet will be correctly decoded. Similarly, if $\delta_H(\mathcal{C}_\alpha) < 0$ for one or more $\mathcal{C}_\alpha \in \Psi$ then a decoding error will be made. If $S_0^{(j_b-1)} = \mathcal{D}_0^{(j_b-1)}$ and there is a tie at position j_b between $\mathcal{C}_0$ and one or more $\mathcal{C}_\alpha$ in $\psi(j_b)$ for the minimum distance codeword from the received sequence, one of the two corresponding survivor paths is chosen at random. Therefore, if $\delta_H(\mathcal{C}_\alpha) \geq 0$ for all $\mathcal{C}_\alpha \in \Psi$ and there are i positions j_b such that for some codeword $\mathcal{C}_\alpha \in \psi(j_b)$, $\delta_H(\mathcal{C}_\alpha) = 0$, the packet is correctly decoded with probability $(1/2)^i$.

In order to simplify the analysis in the case when there is a tie for the minimum distance path at some position j_b, the event that $\mathcal{C}_\alpha$ is favored over $\mathcal{C}_0$, denoted by $\mathcal{F}_\alpha$, is introduced. If $\delta_H(\mathcal{C}_\alpha) > 0$ then with probability one, $\mathcal{C}_\alpha$ is not favored, and if $\delta_H(\mathcal{C}_\alpha) < 0$ then with probability one, $\mathcal{C}_\alpha$ is favored. If $\delta_H(\mathcal{C}_\alpha) = 0$, then $\mathcal{C}_\alpha$ is favored with probability 1/2, where the choice is made independently of all other codewords. The complementary event $\mathcal{F}_\alpha^c$ is the event that $\mathcal{F}_\alpha$ does not occur.

When the received sequence is such that there is not a tie with $\mathcal{C}_0$ for the minimum distance codeword, the event that one or more codewords in $\psi(j_b)$ is favored is equivalent to the event that a decoder error is made. In the case of a tie between $\mathcal{C}_0$ and i codewords in $\psi(j_b)$, the probability that no codeword in $\psi(j_b)$ is favored is $(1/2)^i$, while the probability of no decoder error at position j_b is 1/2.

Thus,

$$P_C(j_b|1,2,\ldots,j_b-1) \geq \Pr(\mathcal{F}_\alpha^c\ \forall\alpha \in \psi(j_b)|\text{ no error at } 1,2,\ldots,j_b-1);$$
$$P_S \geq \Pr(\mathcal{F}_\alpha^c\ \forall\alpha \in \Psi). \tag{5.3}$$

$P_C(j_b)$ is defined as $\Pr(\mathcal{F}_\alpha^c\ \forall\alpha \in \psi(j_b))$, without the condition of no previous errors. In the next section, it is proven that

$$\Pr\left(\mathcal{F}_\beta^c|\mathcal{F}_{\alpha_1}^c,\mathcal{F}_{\alpha_2}^c,\ldots,\mathcal{F}_{\alpha_\mathcal{M}}^c\right) \geq \Pr\left(\mathcal{F}_\beta^c\right). \tag{5.4}$$

for any codeword $\mathcal{C}_\beta$ and any disjoint set of $\mathcal{M}$ codewords $\mathcal{C}_{\alpha_m}$, where the codewords are any of the $2^\mathcal{L}$ codewords of length $\mathcal{L}+K-1$. This proof is valid even when the probability of symbol error varies during the packet reception. Thus,

$$P_C(j_b) \geq \prod_{\alpha\in\psi(j_b)} \Pr(\mathcal{F}_\alpha^c) \qquad \text{and} \qquad P_S \geq \prod_{j_b=K}^{\mathcal{L}+K-1} \prod_{\alpha\in\psi(j_b)} \Pr(\mathcal{F}_\alpha^c). \tag{5.5}$$

5.3 Bound on Conditional Probability

In this section, it is proven that the probability of $\mathcal{F}_\beta^c$ conditioned on the events $\mathcal{F}_{\alpha_1}^c,\mathcal{F}_{\alpha_2}^c,\ldots$, and $\mathcal{F}_{\alpha_\mathcal{M}}^c$ is greater than the unconditional probability of $\mathcal{F}_\beta^c$ for any codeword $\mathcal{C}_\beta$ and any set of $\mathcal{M}$ codewords $\{\mathcal{C}_{\alpha_m}\}$, where the codewords are of length $2(\mathcal{L}+K-1)$ symbols. The inequality is implicitly conditioned on the sequence of channel states for each symbol, $\mathcal{P}$. The following approach is used. First, a set of random variables $\{d_i\}$ is identified as a sufficient statistic for determining the probability of $\mathcal{F}_\beta^c$. Conditioning on the d_i, a recursive expression is found for the

sum over all values of d_i for a given i. By summing over all d_i for every i, a bound on the total probability is found to be

$$\Pr(\mathcal{F}_\beta^c | \mathcal{F}_{\alpha_1}^c, \ldots, \mathcal{F}_{\alpha_\mathcal{M}}^c) \geq \Pr(\mathcal{F}_\beta^c). \tag{5.6}$$

In this section, the word position refers to symbol position, not bit position. For notational convenience, U is used to denote the event $\mathcal{F}_\beta^c$ and V, the intersection of events $\mathcal{F}_{\alpha_1}^c, \mathcal{F}_{\alpha_2}^c, \ldots$, and $\mathcal{F}_{\alpha_\mathcal{M}}^c$.

Consider the ordered collection of codewords C_β, C_{α_1}, $\ldots$,$C_{\alpha_\mathcal{M}}$, each of length $2(\mathcal{L}+K-1)$ symbols. At each position j_s there is a pattern formed by the ordered sequence of symbols $(C_\beta(j_s), C_{\alpha_1}(j_s),\ldots,C_{\alpha_\mathcal{M}}(j_s))$. The symbol i designates the ordered pattern of symbols that is equal to a base 2 representation of i, so $i=0$ is the pattern $(0,0,\ldots,0)$, $i=1$ is $(0,0,\ldots,1)$, and so on up to $i=2^{\mathcal{M}+1}-1$, which is $(1,1,\ldots,1)$. The positions j_s can be grouped according to the patterns. $\mathcal{Y}(i)$ is defined to be the set of all positions j_s for which the pattern is equal to a given i. Some of these sets may be empty. For example, in the case of two codewords C_β and C_α, $\mathcal{Y}(1)$ indicates all positions j_s for which $C_\beta(j_s)=0$ and $C_\alpha(j_s)=1$, $\mathcal{Y}(2)$ those positions for which $C_\beta(j_s)=1$ and $C_\alpha(j_s)=0$, and so forth. $\mathcal{N}_\mathcal{Y}(i)$ denotes the cardinality of the set $\mathcal{Y}(i)$. Furthermore, $\mathcal{Y}_R(i)$ denotes the subset of $\mathcal{Y}(i)$ for which $S_R(j_s)$, the symbol of the received sequence at position j_s, is also equal to 1, and $\mathcal{N}_{\mathcal{Y}_R}(i)$ denotes the cardinality of $\mathcal{Y}_R(i)$.

In evaluating the probability of $\mathcal{F}_\beta^c$, those sets $\mathcal{Y}(i)$ for which the symbol corresponding to the codeword C_β is equal to 1 are of particular interest. For notational convenience, let $n_i = \mathcal{N}_\mathcal{Y}(i+2^\mathcal{M})$, $d_i = \mathcal{N}_{\mathcal{Y}_R}(i+2^\mathcal{M})$, $N = 2^\mathcal{M}-1$, and let ϵ_i refer to a specific value of the random variable d_i, $0 \leq \epsilon_i \leq n_i$. Thus, d_0 is the number of positions for which the received sequence is one, C_β is one, and all of the C_α's are zero; d_N is the number of positions for which all of these are one. The vector $(d_0, d_1, \ldots, d_i)$, is denoted by $\vec{\mathbf{d}}_i$ and similarly for $\vec{\mathbf{e}}_i$.

Given the values $\mathcal{N}_{\mathcal{Y}_R}(i)$, an event $\mathcal{F}^c_{\gamma_1}$ is conditionally independent of both the received sequence $\mathcal{S}_R$ and all other events $\mathcal{F}^c_{\gamma_2}$, where γ_1 and γ_2 can be β or one of the α_m. The numbers $\mathcal{N}_{\mathcal{Y}}(i)$ are determined by the set of codewords $\mathcal{C}_\beta$ and $\{\mathcal{C}_{\alpha_m}\}$. The numbers $\mathcal{N}_{\mathcal{Y}_R}(i)$ are random variables, since they are functions of the random channel errors. Also, the $\mathcal{N}_{\mathcal{Y}_R}(i)$ for different patterns i are independent, since they depend upon the symbol errors occurring during disjoint sets of positions, and the symbol errors are independent conditioned on $\mathcal{P}$. Furthermore, $\delta_H(\mathcal{C}_\beta)$, and consequently $\Pr(\mathcal{F}^c_\beta)$, does not depend upon $S_R(j_s)$ at positions j_s for which $C_\beta(j_s) = 0$, so $\mathcal{F}^c_\beta$ is independent of $\mathcal{N}_{\mathcal{Y}_R}(i)$ for $i < 2^{\mathcal{M}}$. As a result,

$$\Pr(\mathcal{F}^c_\beta | \mathcal{F}^c_{\alpha_1}, \mathcal{F}^c_{\alpha_2}, \ldots, \mathcal{F}^c_{\alpha_{\mathcal{M}}}, \vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N) = \Pr(\mathcal{F}^c_\beta | \vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N). \tag{5.7}$$

Thus,

$$\begin{aligned} &\Pr(\mathcal{F}^c_\beta | \mathcal{F}^c_{\alpha_1}, \mathcal{F}^c_{\alpha_2}, \ldots, \mathcal{F}^c_{\alpha_{\mathcal{M}}}) \\ &\quad = \sum_{e_0=0}^{n_0} \sum_{e_1=0}^{n_1} \cdots \sum_{e_N=0}^{n_N} \Pr(\mathcal{F}^c_\beta | \vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N) \Pr(\vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N | \mathcal{F}^c_{\alpha_1}, \mathcal{F}^c_{\alpha_2}, \ldots, \mathcal{F}^c_{\alpha_{\mathcal{M}}}). \end{aligned} \tag{5.8}$$

As long as the probability of symbol error is always less that 1, the probabilities $\Pr(U)$ and $\Pr(V)$ will be non-zero. For some values of $\vec{\mathbf{d}}_N$, the joint probability $\Pr(\vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N, U, V)$ may equal 0. In such a case, each n_i denotes the largest value of e_i for which the terms of the sum over d_i are non-zero. Thus, in the following, $\Pr(U|\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i)$, $\Pr(V|\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i)$, $\Pr(d_i = e_i)$, and $\Pr(V)$ are non-zero for all $e_i \leq n_i$.

It is first shown that $\Pr(V | d_{i^\bullet} = e_{i^\bullet}, \vec{\mathbf{d}}_{i^\bullet - 1} = \vec{\mathbf{e}}_{i^\bullet - 1})$ is monotonically decreasing with $e_{i^\bullet}$. The probability of the event V is greater than zero if the numbers $\mathcal{N}_{\mathcal{Y}_R}(i)$ satisfy a number of inequalities corresponding to the constraints

$$\delta_H(\mathcal{C}_{\alpha_m}) \geq 0 \qquad m = 1, 2, \ldots, \mathcal{M}. \tag{5.9}$$

These inequalities have the form

$$1/2(\mathcal{N}_Y(i'_1)+\mathcal{N}_Y(i'_2)+\cdots+\mathcal{N}_Y(i'_{\mathcal{Z}})) \geq \mathcal{N}_{Y_R}(i'_1)+\mathcal{N}_{Y_R}(i'_2)+\cdots+\mathcal{N}_{Y_R}(i'_{\mathcal{Z}}), \quad (5.10)$$

where the patterns $i'_1, i'_2, \ldots, i'_{\mathcal{Z}}$ are all those patterns for which the symbol of a given codeword $\mathcal{C}_{\alpha_m}$ equals 1 (so $\mathcal{Z} = 2^{\mathcal{M}}$). If strict inequality holds in all cases then $\Pr(V) = 1$. Otherwise, $0 < \Pr(V) < 1$. If one or more of these strict inequalities is reversed, $\Pr(V) = 0$. If the set $i'_1, i'_2, \ldots, i'_{\mathcal{Z}}$ for a particular inequality does not include $i^* + 2^{\mathcal{M}}$, then d_{i^*} does not affect this inequality. Otherwise, an increase in d_{i^*} causes an increase in the right hand side of the inequality. For a given set of values $\{\mathcal{N}_{Y_R}(i') \; \forall i' \neq i + 2^{\mathcal{M}}\}$, increasing d_{i^*} from e_{i^*} to $e_{i^*} + 1$ can either cause an equality to be true where previously the strict inequality held, or can cause an inequality to be false where previously an equality held, or can have no effect on the veracity of an inequality. In all cases, $\Pr(V)$ either decreases or is unchanged. Thus, summing over all possible values of $\mathcal{N}_{Y_R}(i')$ for all $i' < 2^{\mathcal{M}}$ and $i' > 2^{\mathcal{M}} + i^*$, that is, all $\mathcal{N}_{Y_R}$ not included in the vector $\vec{\mathbf{d}}_{i^*}$, gives

$$\begin{aligned}
&\Pr(V|d_{i^*} = e_{i^*}, \vec{\mathbf{d}}_{i^*-1} = \vec{\mathbf{e}}_{i^*-1}) \\
&= \sum \Pr(V|\mathcal{N}_{Y_R}(0),\ldots,\mathcal{N}_{Y_R}(i'),\ldots,\mathcal{N}_{Y_R}(2^{\mathcal{M}+1}-1), \vec{\mathbf{d}}_{i^*-1}, d_{i^*} = e_{i^*}) \\
&\qquad \times \Pr(\mathcal{N}_{Y_R}(0),\ldots,\mathcal{N}_{Y_R}(i'),\ldots,\mathcal{N}_{Y_R}(2^{\mathcal{M}+1}-1)|\vec{\mathbf{d}}_{i^*-1}, d_{i^*} = e_{i^*}) \\
&\geq \sum \Pr(V|\mathcal{N}_{Y_R}(0),\ldots,\mathcal{N}_{Y_R}(i'),\ldots,\mathcal{N}_{Y_R}(2^{\mathcal{M}+1}-1), \vec{\mathbf{d}}_{i^*-1}, d_{i^*} = e_{i^*}+1) \\
&\qquad \times \Pr(\mathcal{N}_{Y_R}(0),\ldots,\mathcal{N}_{Y_R}(i'),\ldots,\mathcal{N}_{Y_R}(2^{\mathcal{M}+1}-1)|\vec{\mathbf{d}}_{i^*-1}, d_{i^*} = e_{i^*}+1) \\
&\geq \Pr(V|d_{i^*} = e_{i^*}+1, \vec{\mathbf{d}}_{i^*-1} = \vec{\mathbf{e}}_{i^*-1}). \qquad (5.11)
\end{aligned}$$

Thus, for any index i, dividing both sides by $\Pr(V)$ gives

$$\begin{aligned}
\frac{\Pr(V, d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})}{\Pr(d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Pr(V)} &\geq \frac{\Pr(V, d_i = e_i+1, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})}{\Pr(d_i = e_i+1, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Pr(V)} \\
\frac{\Pr(d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}|V)}{\Pr(d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})} &\geq \frac{\Pr(d_i = e_i+1, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}|V)}{\Pr(d_i = e_i+1, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})}. \qquad (5.12)
\end{aligned}$$

Since this ratio is monotonically decreasing with e_i, there are three possibilities

$$\frac{\Pr(d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}|V)}{\Pr(d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})} \begin{cases} \geq 1, & \text{for all } e_i,\ 0 \leq e_i \leq n_i; \\ < 1, & \text{for all } e_i,\ 0 \leq e_i \leq n_i; \\ \text{crosses } 1, & \text{for some } c_i,\ 0 \leq c_i \leq n_i, \end{cases} \tag{5.13}$$

where "crosses 1" is used to mean that it is greater than or equal to one for $e_i \leq c_i$ and is less than one for $e_i > c_i$. $\Delta_i(\vec{\mathbf{e}}_i)$ is defined as

$$\Delta_i(\vec{\mathbf{e}}_i) = \Pr(\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i|V) - \Pr(\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i). \tag{5.14}$$

Because the probabilities are non-negative, corresponding inequalities for $\Delta_i(\vec{\mathbf{e}}_i)$ can be found.

$$\Delta_i(\vec{\mathbf{e}}_i) \begin{cases} \geq 0, & \text{for all } e_i,\ 0 \leq e_i \leq n_i; \\ < 0, & \text{for all } e_i,\ 0 \leq e_i \leq n_i; \\ \text{crosses } 0, & \text{for some } c_i,\ 0 \leq e_i \leq n_i. \end{cases} \tag{5.15}$$

Thus, either there is an integer c_i in the range $0 \leq c_i \leq n_i$ such that $\Delta_i(\vec{\mathbf{e}}_i) \geq 0$ for $e_i \leq c_i$ and $\Delta_i(\vec{\mathbf{e}}_i) < 0$ for $e_i > c_i$, or else $\Delta_i(\vec{\mathbf{e}}_i) < 0$ for all e_i. From Eqn. 5.8,

$$\Pr(U|V) - \Pr(U) = \sum_{\vec{\mathbf{e}}_N} \Pr(U|\vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N)\Big(\Pr(\vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N|V) - \Pr(\vec{\mathbf{d}}_N = \vec{\mathbf{e}}_N)\Big). \tag{5.16}$$

In order to derive a recursion expression, the general form of a single summation of Eqn. 5.16 is considered,

$$\sum_{\vec{e}_i} \Pr(U|\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i)\Big(\Pr(\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i|V) - \Pr(\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i)\Big) = \sum_{\vec{e}_i} \Pr(U|\vec{\mathbf{d}}_i = \vec{\mathbf{e}}_i)\Delta_i(\vec{\mathbf{e}}_i). \tag{5.17}$$

In the case $\Delta_i(\vec{\mathbf{e}}_i) \geq 0$ for some $e_i \in (0, 1, \ldots, n_i)$, this sum is split into two parts,

$$\sum_{\vec{e}_i} \Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i) = \sum_{e_i=0}^{c_i} \Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i) + \sum_{e_i=c_i+1}^{n_i} \Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i). \tag{5.18}$$

Eqn. 5.11 also holds for U, so for $e_i \leq c_i$,

$$\Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}) \geq \Pr(U|d_i = c_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}). \tag{5.19}$$

For $e_i \leq c_i$, $\Delta_i(\vec{\mathbf{e}}_i) \geq 0$ by definition. Thus,

$$\Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i) \geq \Pr(U|d_i = c_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i). \tag{5.20}$$

Similarly, for $e_i > c_i$,

$$\Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}) \leq \Pr(U|d_i = c_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}), \tag{5.21}$$

and $\Delta_i(\vec{\mathbf{e}}_i) < 0$, so the inequality reverses to the desired direction. Consequently, both parts of the sum give the inequality of Eqn. 5.20, so

$$\begin{aligned}\sum_{e_i=0}^{n_i} \Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i) &\geq \sum_{e_i=0}^{n_i} \Pr(U|d_i = c_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i) \\ &\geq \Pr(U|d_i = c_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}) \sum_{e_i=0}^{n_i} \Delta_i(\vec{\mathbf{e}}_i).\end{aligned} \tag{5.22}$$

In the case $\Delta_i(\vec{\mathbf{e}}_i) < 0$ for all $e_i \in (0, 1, \ldots, n_i)$, Eqns. 5.20 and 5.22 hold for all e_i if c_i is chosen to be 0. The sum over e_i of $\Delta_i(\vec{\mathbf{e}}_i)$ is found from

$$\begin{aligned}\sum_{e_i=0}^{n_i} \Delta_i(\vec{\mathbf{e}}_i) &= \sum_{e_i=0}^{n_i} \Pr(d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}|V) - \sum_{e_i=0}^{n_i} \Pr(d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}) \\ &= \Pr(\vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}|V) - \Pr(\vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1}) \\ &= \Delta_{i-1}(\vec{\mathbf{e}}_{i-1}).\end{aligned} \tag{5.23}$$

Thus,

$$\sum_{e_i} \Pr(U|d_i = e_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_i(\vec{\mathbf{e}}_i) \geq \Pr(U|d_i = c_i, \vec{\mathbf{d}}_{i-1} = \vec{\mathbf{e}}_{i-1})\Delta_{i-1}(\vec{\mathbf{e}}_{i-1}). \tag{5.24}$$

Now, Eqn. 5.16 can be evaluated as

$$\begin{aligned}
\Pr(U|V) - \Pr(U) &= \sum_{\vec{e}_{N-1}} \sum_{e_N} \Pr(U|d_N = e_N, \vec{\mathbf{d}}_{N-1} = \vec{\mathbf{e}}_{N-1}) \Delta_N(\vec{\mathbf{e}}_N) \\
&\geq \sum_{\vec{e}_{N-1}} \Pr(U|d_N = c_N, \vec{\mathbf{d}}_{N-1} = \vec{\mathbf{e}}_{N-1}) \Delta_{N-1}(\vec{\mathbf{e}}_{N-1}) \\
&\geq \sum_{\vec{e}_{N-2}} \Pr(U|d_N = c_N, d_{N-1} = c_{N-1}, \vec{\mathbf{d}}_{N-2} = \vec{\mathbf{e}}_{N-2}) \Delta_{N-2}(\vec{\mathbf{e}}_{N-2}) \\
&\quad \vdots \\
&\geq \sum_{e_0} \Pr(U|\vec{\mathbf{d}}_N = \vec{\mathbf{c}}_N) \Delta_0(e_0) \\
&\geq \Pr(U|\vec{\mathbf{d}}_N = \vec{\mathbf{c}}_N) \sum_{e_0} \Big(\Pr(d_0 = e_0|V) - \Pr(d_0 = e_0) \Big). \qquad (5.25)
\end{aligned}$$

However,

$$\sum_{e_0} \Pr(d_0 = e_0|V) = \sum_{e_0} \Pr(d_0 = e_0) = 1, \qquad (5.26)$$

so the difference in the last line of Eqn. 5.25 is zero, giving $\Pr(U|V) - \Pr(U) \geq 0$, or

$$\Pr(\mathcal{F}_\beta^c | \mathcal{F}_{\alpha_1}^c, \ldots, \mathcal{F}_{\alpha_\mathcal{M}}^c) \geq \Pr(\mathcal{F}_\beta^c). \qquad (5.27)$$

5.4 Numerical Evaluation of the Probability of Correct Decoding

The product given in Eqn. 5.5 is evaluated in this section. The sets $\psi(j_b)$ are all subsets of a shifted version of $\psi(\mathcal{L} + K - 1)$. The set ψ^* is defined to be the set of all codewords of length $\mathcal{L} + K - 1$ which first return to the zero state at position $j = \mathcal{L} + K - 1$, namely $\psi(\mathcal{L} + K - 1)$. To find the bound on $P_C(j_{b_0})$, each codeword

$C_\alpha \in \psi(j_{b_0})$ is mapped into $C_{\alpha'} \in \psi^*$ as follows. If $j'_s = j_s + 2(\mathcal{L}+K-1-j_{b_0})$, $C_{\alpha'}$ is the codeword for which

$$C_{\alpha'}(j'_s) = \begin{cases} 0, & j'_s \leq 2(\mathcal{L}+K-1-j_{b_0}); \\ C_\alpha(j'_s - 2(\mathcal{L}+K-1-j_{b_0})), & j'_s > 2(\mathcal{L}+K-1-j_{b_0}). \end{cases} \tag{5.28}$$

$P_C(j_{b_0})$ is lower bounded by the probability of not favoring any codewords in ψ^*, given a probability of symbol error $p_{j'_s} = 0$ for $j'_s = 1$ to $2(\mathcal{L}+K-1-j_{b_0})$, and $p_{j'_s} = p_{j_s}$ for $j'_s = 2(\mathcal{L}+K-j_{b_0})-1$ to $j'_s = 2(\mathcal{L}+K-1)$ (i.e., a shift of $\mathcal{L}+K-1-j_{b_0}$ bits or $2(\mathcal{L}+K-1-j_{b_0})$ symbols). With this bound, the bit position j_b is only needed to determine the probabilities of symbol error at each symbol position j'_s, so $P^*_C(p_1, p_2, \ldots, p_{2(\mathcal{L}+K-1)})$ will be used to indicate the bound on $P_C(j_b)$ with the given pattern of symbol error probabilities. This is a slight shift of notation, as P^*_C is a function of the probabilities of symbol error, while P_C is a function of position.

An empirical result leads to a reduction in the number of codewords that must be considered. It has been found that with very high probability, the survivor paths merge within 4 or 5 constraint lengths in the past [30], [10]. In other words, at any position j_b, the survivor paths $S_{\mathcal{E}}^{(j_b)}$ for all states $\mathcal{E}$ will be equal from positions 1 to $j_b - i$, where i is a random variable that depends upon the error pattern occurring in the received sequence, but is less than 5 constraint lengths, or 35 bits, with very high probability. Therefore, if the all zero path is replaced as a survivor at position j_b by some path $\mathcal{D}_\alpha^{(j_b)}$, then with very high probability this path is identical to $\mathcal{D}_0$ for all positions prior to $j_b - 34$. Thus ψ^* can be limited to those codewords which are in the zero state at all positions except for positions j'_b from $\mathcal{L}+K-35$ to $\mathcal{L}+K-1$.

$P_C(j_b)$ is a monotonic function of the probability of symbol error. Therefore, the bound on this probability can be bracketed by considering the maximum and minimum probability of symbol error to occur during the interval from bit positions

$j_b' = \mathcal{L} + K - 35$ to $j_b' = \mathcal{L} + K - 1$. This reduces the problem to one of evaluating $P_C^*(p_1, \ldots, p_{2(\mathcal{L}+K-1)})$ with a constant probability of symbol error p. For simplicity, this is denoted by $P_C^*(p)$.

$P_C^*(p)$ is found by considering the weight structure of the code being used. For a constant probability of symbol error p, the probability that a codeword C_α of weight i is favored is equal to $P_i(p)$, which is simply the probability that more than half of the i non-zero symbols of C_α are in error [51].

$$\Pr(\mathcal{F}_\alpha) = P_i(p) = \begin{cases} \displaystyle\sum_{e=(i+1)/2}^{i} \binom{i}{e} p^e (1-p)^{i-e}, & i \text{ odd;} \\ \displaystyle\frac{1}{2}\binom{i}{i/2} p^{i/2}(1-p)^{i/2} + \sum_{e=i/2+1}^{i} \binom{i}{e} p^e (1-p)^{i-e}, & i \text{ even.} \end{cases} \tag{5.29}$$

Let a_i^j be the number of codewords with a single return to zero which are of weight i and which spend a total of j or fewer bits out of the zero state, as in [51]. The number of codewords in ψ^* of weight i is equal to a_i^{35}. Also, there are at most 70 symbols during which a codeword in ψ^* can be non-zero, so $a_i^{35} = 0$ for $i > 70$. For a constant symbol error rate p, Eqn. 5.5 reduces to

$$P_C^*(p) = \prod_{i=1}^{70} (1 - P_i(p))^{a_i^{35}}. \tag{5.30}$$

As described in [51], a_i^j, can be found by representing the encoder as a signal flow graph and finding the transfer function $T(D, L, N)$ of this graph. Using the example of the $K = 3$ code, $T(D, L, N)$ is found by modifying the state diagram of Fig. 5.1 as follows. Split the zero node into a source node and a sink node. Assign a gain of $LN^\beta D^\gamma$ to the transitions, where β is the weight of the input bit (i.e., 0 for a 0, 1 for a 1), and γ is the weight of the output bit (i.e., 0 for 00, 1 for 01 or 10, and 2 for 11). For a path (codeword) of length α' with β' ones in the input bits and γ'

ones in the output symbols, the product of the transition gains is $L^{\alpha'}N^{\beta'}D^{\gamma'}$. The modified state diagram is given in Fig. 5.2. Summing together all possible paths from the source to the sink will give the overall gain of the network, $T(D, L, N)$. For example, for the rate 1/2, $K = 3$ code, this is

$$T(D, L, N) = D^5L^3N + D^6L^4(1 + L)N^2 + ... + D^{5+j}L^{3+j}(1 + L)^jN^{1+j} + .. \quad (5.31)$$

This gain can be found directly from the modified state diagram by solving the set of linear equations giving the dependencies between the nodes. In the simple example, $T(D, L, N)$ is found directly as

$$T(D, L, N) = \frac{D^5L^3N}{1 - DL(1 + L)N}. \quad (5.32)$$

Because a single bit error and many bit errors have the same effect of causing the packet to be unsuccessful, N, the number of ones in the input bits, is irrelevant. Setting N equal to 1 gives

$$T(D, L) = \frac{D^5L^3}{1 - DL(1 + L)}. \quad (5.33)$$

The number a_i^j is equal to the sum of the coefficients of all terms of $T(D, L)$ for which the exponent of D is i and the exponent of L is less than or equal to j. Unfortunately, the calculation of $T(D, L)$ is very difficult for the higher constraint length codes. However, a_i^{35} is bounded by $a_i = \lim_{j\to\infty} a_i^j$, and the terms a_i can be found from the simpler transfer function $T(D)$, defined as

$$T(D) = T(D, L)|_{L=1}. \quad (5.34)$$

If a series expansion for $T(D)$ is known, the terms a_i can be found from

$$T(D) = \sum_{i=1}^{\infty} a_i D^i. \quad (5.35)$$

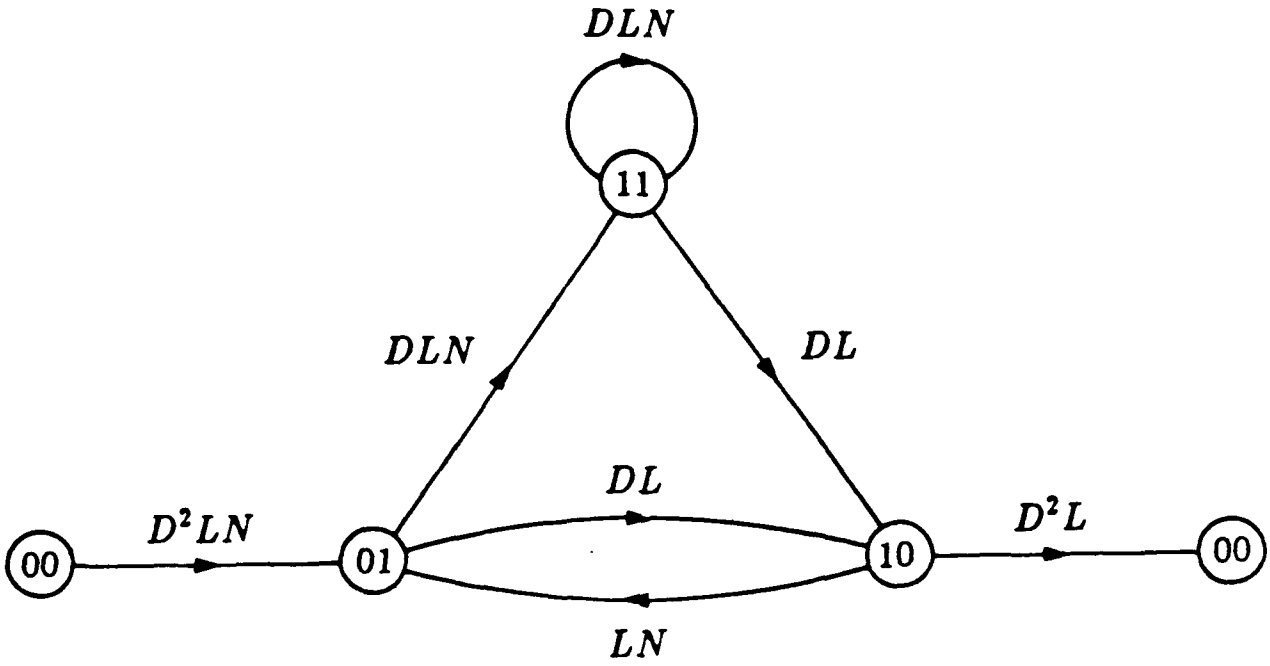


Fig. 5.2 Modified state diagram. Rate 1/2, K=3 code.

Typically, for larger constraint lengths, such as the $K = 7$ code, the first few coefficients a_i are found by inverting the numerical matrix resulting for a specific value of D, and fitting a polynomial to the resulting function $T(D)$. However, this method would require extreme accuracy to find a_i for moderate values of i.

By solving* the set of linear equations consisting of polynomials of D, an exact expression is found for $T(D)$. The expression is a ratio of polynomials, whose numerator and denominator are given by

$$\text{Numerator}=11D^{10} - 6D^{12} - 25D^{14} + D^{16} + 93D^{18} - 15D^{20} - 176D^{22} - 76D^{24} + 243D^{26} + 417D^{28} - 228D^{30} - 1156D^{32} - 49D^{34} + 2795D^{36} + 611D^{38} - 5841D^{40} - 1094D^{42} + 9575D^{44} + 1097D^{46} - 11900D^{48} - 678D^{50} + 11218D^{52} + 235D^{54} - 8068D^{56} - 18D^{58} + 4429D^{60} - 20D^{62} - 1838D^{64} + 8D^{66} + 562D^{68} - D^{70} - 120D^{72} + 16D^{76} - D^{80}$$

$$\text{Denominator}=1 - 4D^2 - 6D^4 - 30D^6 + 40D^8 + 85D^{10} - 81D^{12} - 345D^{14} + 262D^{16} + 844D^{18} - 403D^{20} - 1601D^{22} + 267D^{24} + 2509D^{26} + 389D^{28} - 3064D^{30} - 2751D^{32} + 2807D^{34} + 8344D^{36} - 1960D^{38} - 16133D^{40} + 1184D^{42} + 21746D^{44} - 782D^{46} - 21403D^{48} + 561D^{50} + 15763D^{52} - 331D^{54} - 8766D^{56} + 131D^{58} + 3662D^{60} - 30D^{62} - 1123D^{64} + 3D^{66} + 240D^{68} - 32D^{72} + 2D^{76}$$

The fraction is expanded by using the identity

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots \tag{5.36}$$

Each term of this expansion is itself a polynomial of D. The exact coefficients are calculated out to D^{60}. Multiplying by the numerator gives the exact coefficients a_i of $T(D) = \sum a_i D^i$ for $i \leq 70$. The non-zero coefficients to ten digits accuracy are listed below.

---

*This is accomplished by using the MACSYMA (©1976, 1983 Massachusetts Institute of Technology. ©1983 Symbolics, Inc.) symbolic manipulation program.

Coefficients a_i

i	a_i	i	a_i	i	a_i
10	11	32	1409277901	54	$2.912797332 \times 10^{17}$
12	38	34	8034996288	56	$1.660510362 \times 10^{18}$
14	193	36	$4.580875611 \times 10^{10}$	58	$9.466139591 \times 10^{18}$
16	1331	38	$2.611287754 \times 10^{11}$	60	$5.396401324 \times 10^{19}$
18	7275	40	$1.488634502 \times 10^{12}$	62	$3.076348764 \times 10^{20}$
20	40406	42	$8.486419243 \times 10^{12}$	64	$1.753746820 \times 10^{21}$
22	234969	44	$4.837861791 \times 10^{13}$	66	$9.997656840 \times 10^{21}$
24	1337714	46	$2.757937903 \times 10^{14}$	68	$5.699405471 \times 10^{22}$
26	7594819	48	$1.572231420 \times 10^{15}$	70	$3.249083581 \times 10^{23}$
28	43375588	50	$8.962880896 \times 10^{15}$		
30	247339453	52	$5.109505443 \times 10^{16}$		

Given these coefficients a_i, $P_C^*(p)$ can be bounded by $\hat{P}_C(p)$, where

$$P_C^*(p) \geq \hat{P}_C(p) = \prod_{i=1}^{70} (1 - P_i(p))^{a_i}. \tag{5.37}$$

5.5 Memoryless Approximation

The overall bound on the probability of success is denoted by $\hat{P}_S$. Also, $p_{j_s}^+$ is the maximum probability of symbol error occurring during symbol positions $j_s - 69$ to j_s, and $p_{j_s}^-$ is the minimum. Because $j_s = 2j_b$,

$$\prod_{j_b=K}^{\mathcal{L}+K-1} \hat{P}_C(p_{2j_b}^-) \geq \hat{P}_S \geq \prod_{j_b=K}^{\mathcal{L}+K-1} \hat{P}_C(p_{2j_b}^+). \tag{5.38}$$

A true lower bound on P_S is found by using $p_{j_b}^+$. However, a very complicated network model would be required to keep track of this maximum. Also, it can be shown that although the channel state may vary widely during a packet of 1000 to 4000 bits, the state will be nearly constant over the decoder memory of 35 bits. The time between network state transitions is exponentially distributed. The rate is equal to the sum of the rates of the transitions leaving a given state. For the infinite population model, this is $\Lambda + X\mu$. Normalizing to μ gives a rate of $G + X$. Even for X as large as 50 and G as large as 50, the maximum rate is less than 100. For a mean packet size of 1000 bits, this rate corresponds to a mean holding time of 10 bits. For the finite population model, the rate is $(M - X - R)\lambda + X\mu$. Also, from the numerical results it is seen that the maximum throughput is almost always achieved with a traffic rate g^* less than or nearly equal to 1.0, so for every state (X, R), the transition rate is bounded by $M\mu$. For a mean packet length $1/\mu = 1000$ bits, this gives a mean holding time of at least 33 bits for $M = 30$. Furthermore, when the performance is limited by channel errors, which is the only case for which the approximations have any effect, the throughput is maximized by a $g^* << 1.0$. For g in this range, the mean holding time is close to or greater than about 35 bits for all states with a large steady state probability, even for networks as large as $M = 60$. This suggests that a good approximation to the lower bound on P_S is

$$\hat{P}_S \approx \prod_{j_b=K}^{\mathcal{L}+K-1} \hat{P}_C(p_{2j_b}) \approx \prod_{j_b=1}^{\mathcal{L}} \hat{P}_C(p_{2j_b}). \tag{5.39}$$

where p_{2j_b} is the current value of the probability of symbol error at bit position j_b. For a mean packet length of 1000 bits, the effect of the offset of $K - 1$ bits is negligible, so the packet can be assumed to be of total length $\mathcal{L}$, and the product can be taken over bits $j_b = 1$ to $\mathcal{L}$. The result of Eqn. 5.39 is both memoryless and independent from bit to bit, so the error process can be modeled as a time

varying Bernoulli process for which the parameter at each position is $\hat{P}_C(p_{2j_b})$. The simulation described in Appendix B was used to verify this approximation. As discussed in the next section, the error in the capacity C was found to be less than 5%.

5.6 Coded Channel Results

For comparison to the results for the uncoded channel given in Chapter 4, the received power is stated in terms of E_b/N_0. The symbol energy is simply $E_s/N_0 = 1/2(E_b/N_0)$ for the rate 1/2 code used. Again, results are for a mean packet length $1/\mu$ of 1000 bits.

In Fig. 5.3, the throughput and probability of success are plotted for the infinite population model with the coded channel, $E_b/N_0 = 8.0$ dB, $N = 32$. The throughput is similar to the plot for the uncoded channel, but the maximum is achieved at G^* which gives a much higher probability of success. As discussed in §4.1.4, the coded channel does better under the criterion of maximum normalized throughput achievable with a constraint of a minimum value of P_S. This is shown in Table 5.1, which compares the coded narrowband ($N = 1$) and SSMA ($N = 32$) channels. The narrowband coded channel is penalized by the code rate of 1/2, and so does worse than the uncoded channel. Even with the normalization by W, the SSMA channel clearly outperforms narrowband channel.

The normalized throughput versus normalized traffic is plotted for the infinite population model in Fig. 5.4. This is similar to the plot for the uncoded channel, but for the same spreading factor W, a higher capacity is achieved with the coded channel. Furthermore, the received power of $E_b/N_0 = 8.0$ dB is lower than the 10.0 dB used for the uncoded channel results.

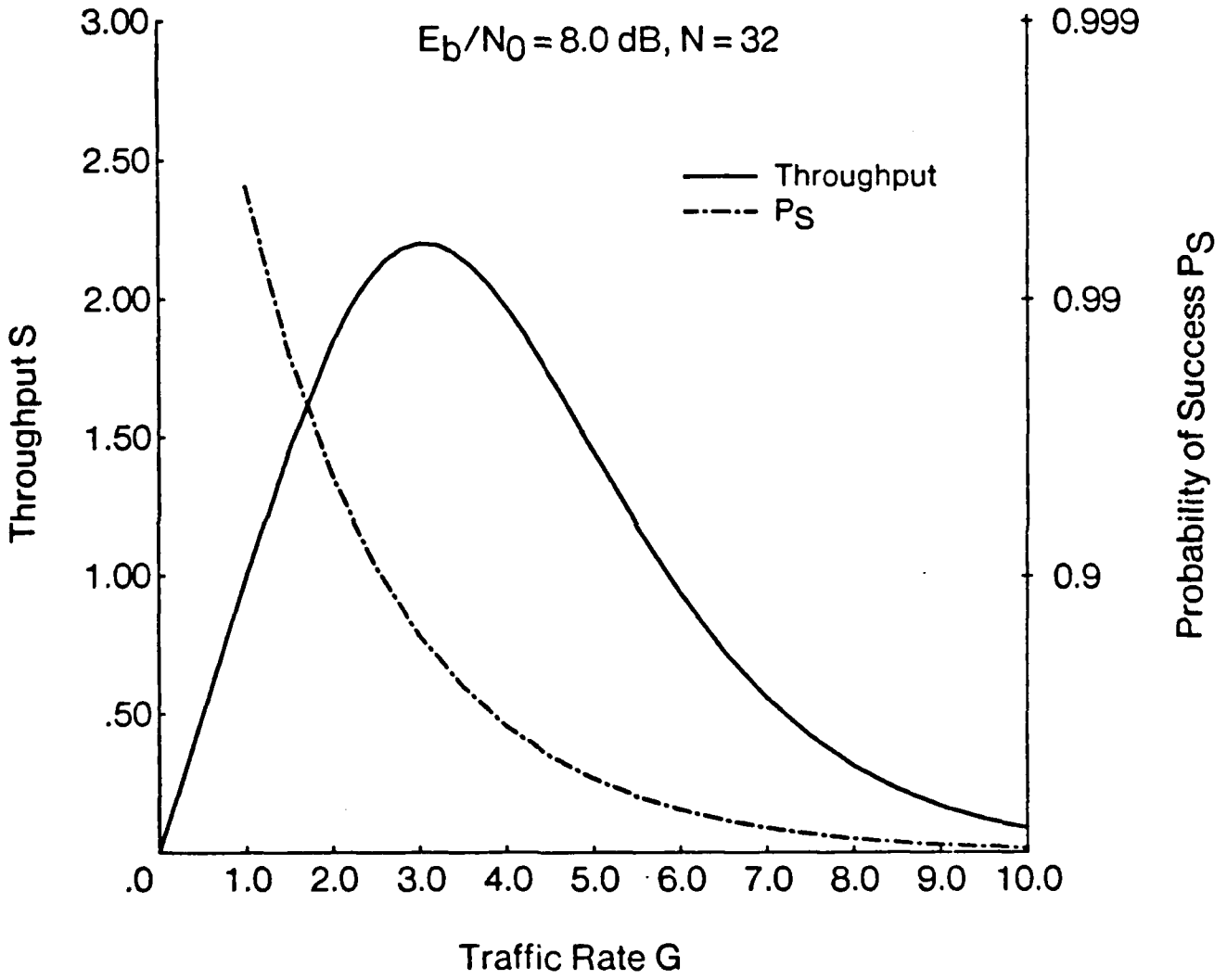


Fig. 5.3 Throughput and probability of success versus traffic rate, infinite population, coded channel.

Minimum P_S	Narrowband	SSMA
0.90	0.0229	0.0331
0.95	0.0120	0.0296
0.99	0.00248	0.0202

Table 5.1. Maximum throughput with constrained P_S, coded narrowband and SSMA channels.

The capacity C and normalized capacity C/W are plotted as a function of the chips per symbol N ($W = 2N$) for several values of E_b/N_0 in Figures 5.5 and 5.6, respectively. Again, these plots are similar to the results for the uncoded channel, but the capacities are higher.

Results for the performance of the finite population network with a coded channel are given in the next figures. Fig. 5.7 shows the capacity as a function of offered traffic for $E_b/N_0 = 8.0$ dB, and for several values of N. Also shown are the results from simulation which give the throughput found using p_j^+, p_j^-, and p_j, as in Eqn. 5.38, for the $N = 32$, $N = 64$, and $N = 256$ curves. The 95% confidence interval for each of these points has a width of 2% to 3% of the values found, indicated by a small vertical line through each point. When using p_j, the simulation is modeling the exact same case as modeled by the analysis. As hoped, the analytical results are within the confidence intervals for all these points. Furthermore, the analytical results are within 5% of the upper and lower bounds, and the results are much closer near the peak. In the case $N = 256$, the channel errors are negligible, so there is virtually no difference between the three cases simulated.

The capacity C is plotted as a function of the number of chips per bit N for several values of E_b/N_0 in Fig. 5.8. Several effects are indicated by these curves. At

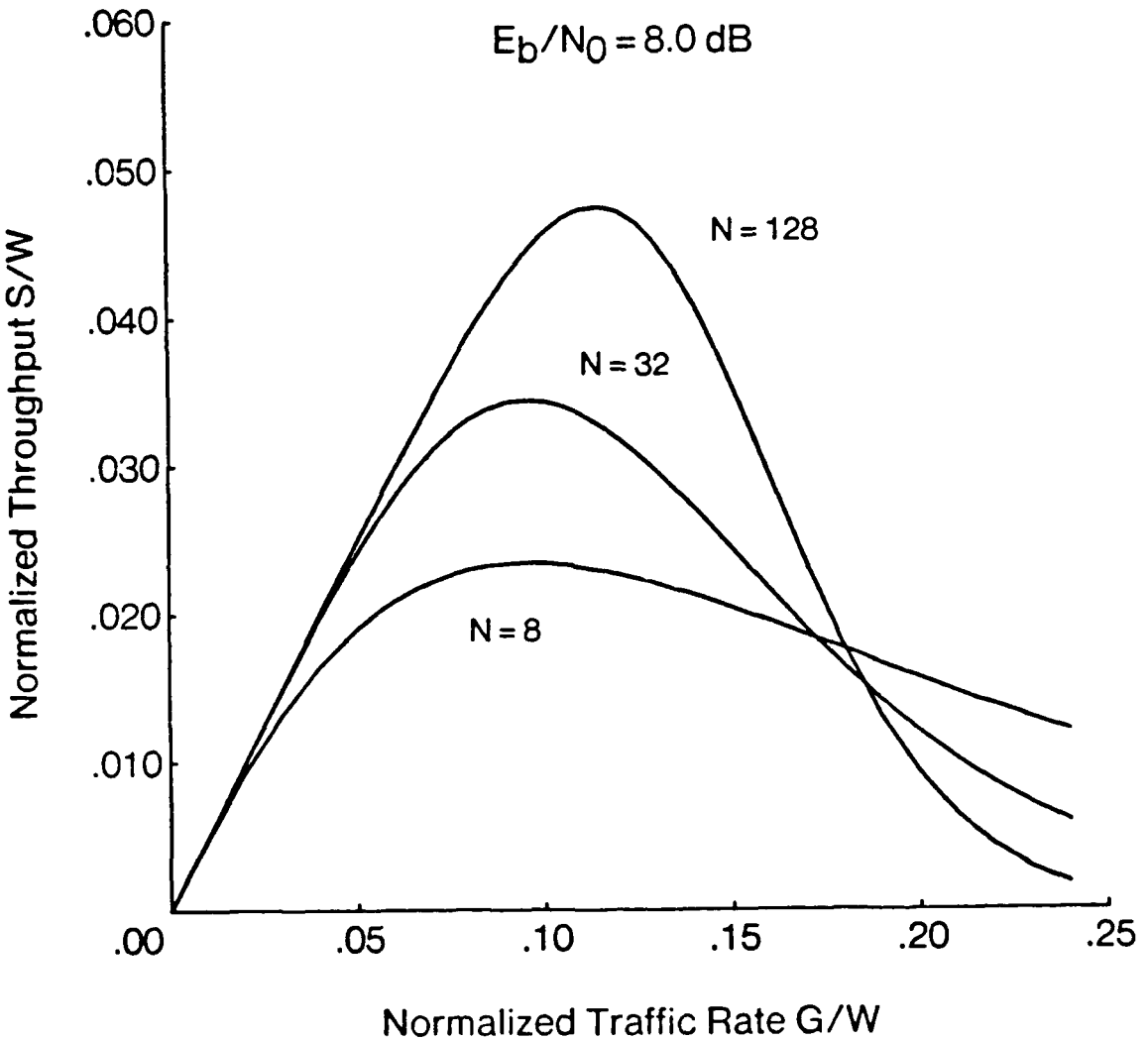


Fig. 5.4 Normalized throughput versus normalized traffic rate, infinite population, coded channel.

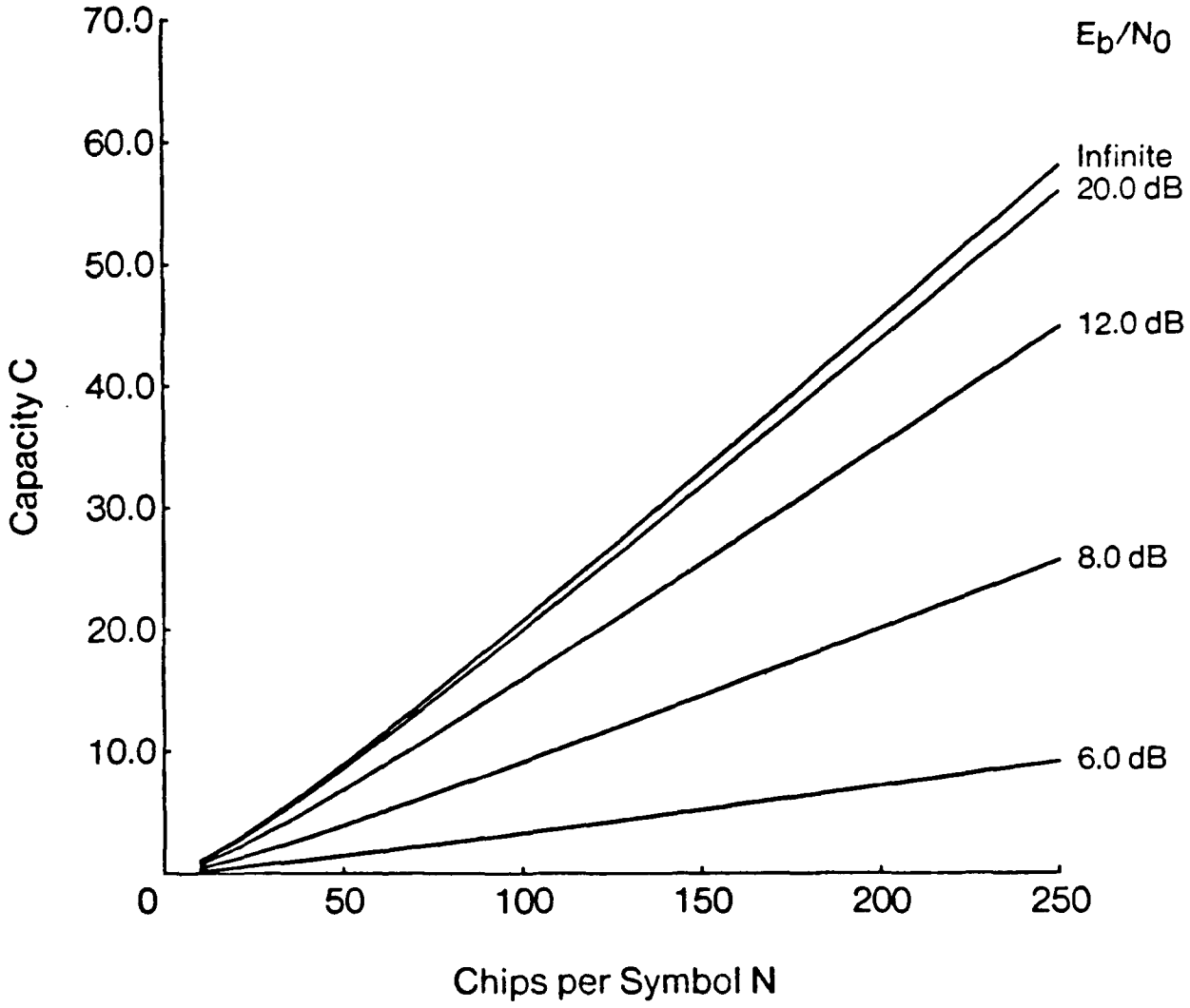


Fig. 5.5 Capacity versus spreading factor and received power, infinite population, coded channel.

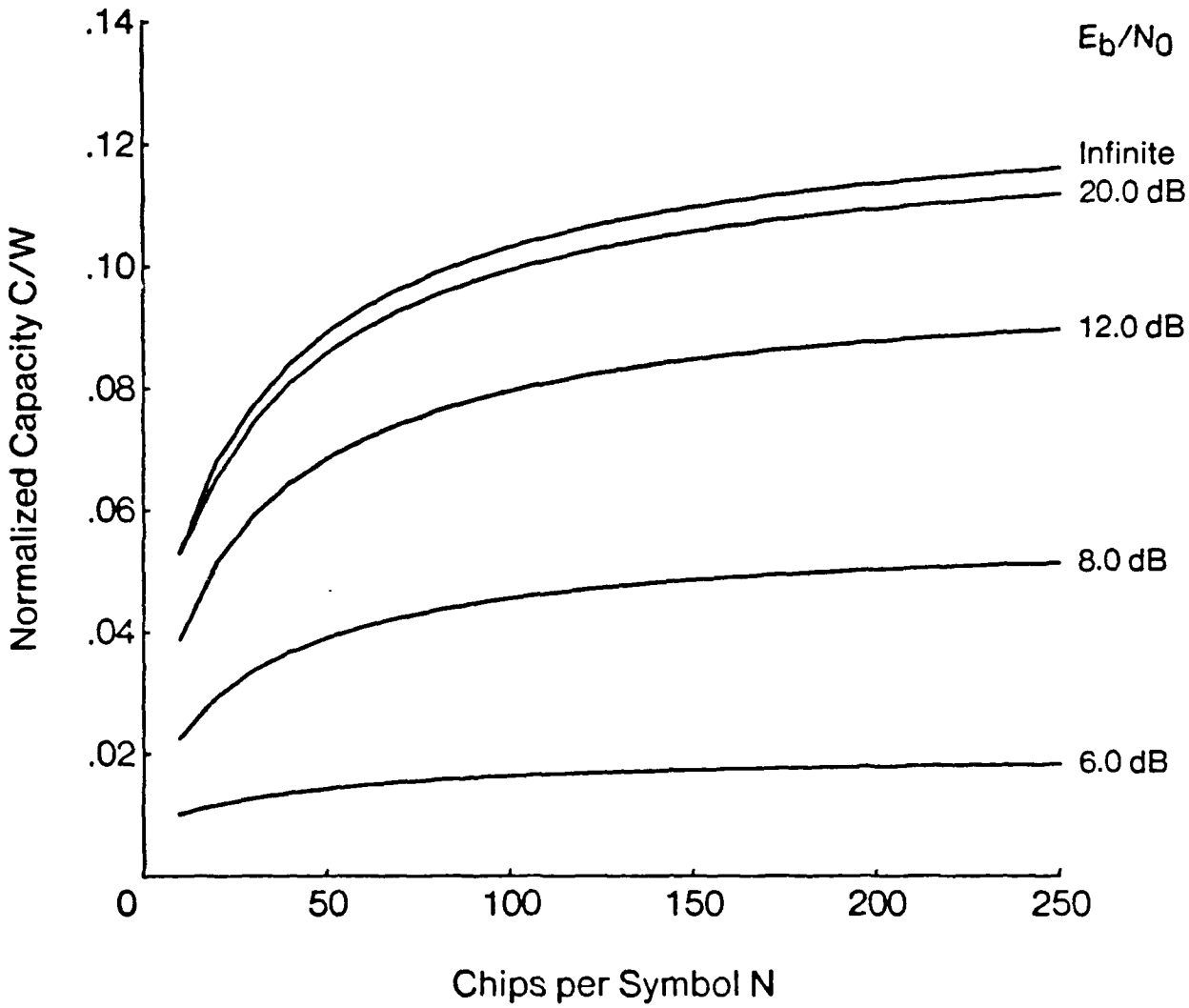


Fig. 5.6 Normalized capacity versus spreading factor and received power, infinite population, coded channel.

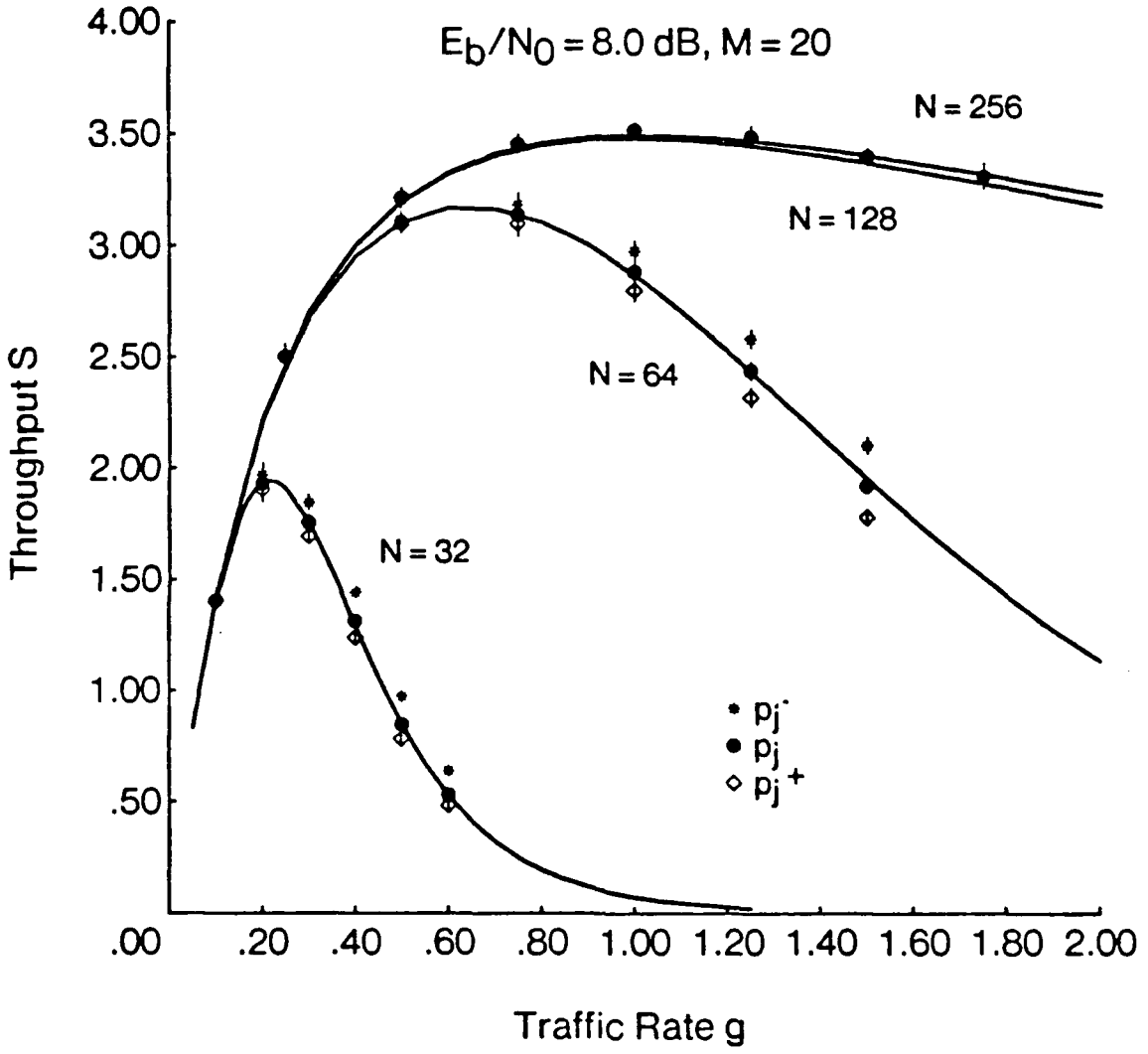


Fig. 5.7 Throughput versus traffic rate, finite population, coded channel.

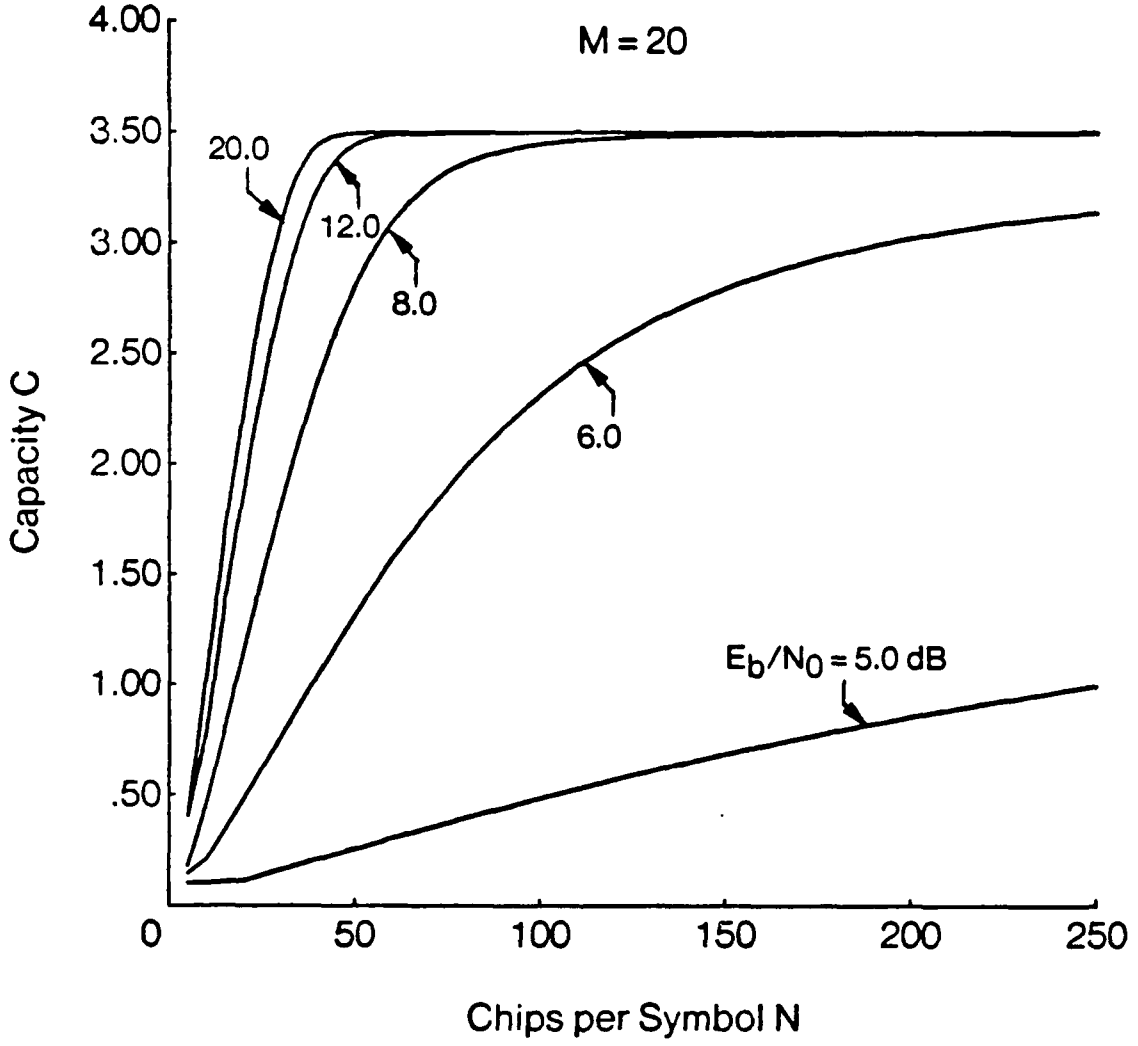


Fig. 5.8 Capacity versus spreading factor and received signal power, finite population, coded channel.

low values of N, capacity is limited by the combination of interference and thermal noise, i.e., the channel errors. At higher N, the effect of interference becomes small. Indeed, because the network size, and thus X, is finite, in the limit as $N \rightarrow \infty$, the symbol error rate for all values of X will approach $P_{e,symbol}(1)$, the performance with no interference for the given E_b/N_0. This asymptotic capacity was calculated, and found to be 2.10 for $E_b/N_0 = 5.0$ dB, and 3.41 for $E_b/N_0 = 6.0$ dB. For $E_b/N_0 \geq 8.0$, with large N, the capacity is limited by the receiver availability rather than by the effect of channel errors, so the asymptotic capacity is equal to the same value of 3.50 for all $E_b/N_0 \geq 8.0$ dB. Furthermore, this value is reached at moderate values of N, as can be seen in the figure.

There appears to be only a small difference between the curves for $E_b/N_0 \geq 8.0$ dB. However, for a fixed value of N which is below the knee of the curves, there actually is a significant improvement for higher E_b/N_0. This effect is hidden by the steepness of the curves. For example, at $N = 32$, the capacity for $E_b/N_0 = 8.0$ dB is 1.94, while the capacity for $E_b/N_0 = 12.0$ dB is 2.86. This becomes more significant for larger network sizes M, for which the knee of the curve occurs at higher values of N. In this case, it is quite conceivable that system constraints would limit N to values below this knee, so the performance can be improved by increasing the power to give E_b/N_0 well above 8.0 dB. Of course, for very large E_b/N_0, the thermal noise is negligible, so the performance for $E_b/N_0 = 20.0$ dB is almost identical to the performance for the case of no thermal noise.

Fig. 5.9 shows the improvement due to FEC coding. The capacity C versus E_b/N_0 is plotted for the coded channel with 32 and 256 chips per symbol, and for the uncoded channel with 64 and 512 chips ber bit. The different spreading factors were chosen to give the same overall bandwidth expansion W. At the higher spreading factors, the use of coding gives a power savings of about 3.0 dB. At lower

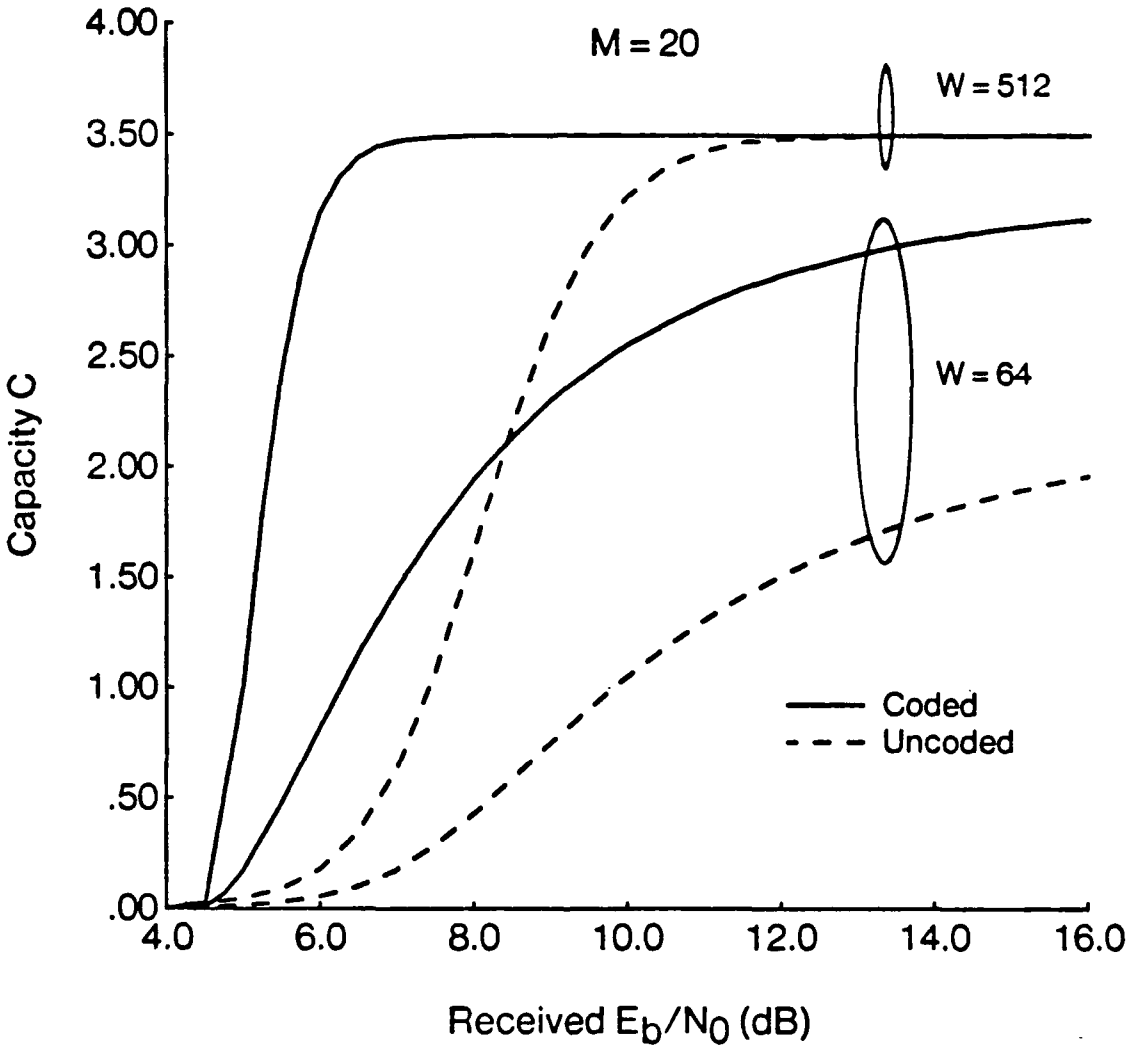


Fig. 5.9 Capacity versus received signal power with and without FEC coding.

spreading factors, even with 3.0 dB higher power, the uncoded channel cannot support as high a throughput as the coded channel. For the limiting case when the thermal noise density is zero ($E_b/N_0 = \infty$), with $W = 64$, the capacity is 2.23 for the uncoded channel and is 3.25 for the coded channel. Therefore, as expected, the use of FEC coding results in a significant improvement in performance, except when the performance of the uncoded system is limited by receiver availability.

The effect of receiver availability is shown explicitly in Fig. 5.10 for the coded channel, with $E_b/N_0 = 8.0$ dB. Results are similar to those for the uncoded channel. However, for the coded channel, the limiting value is reached at a lower N and the plateau is flatter. Thus, the capacity for $W = 500$ is higher for the coded channel with $E_b/N_0 = 8.0$ dB than for the uncoded channel with $E_b/N_0 = 10.0$ dB.

In Fig. 5.11, the network capacity C and user capacity C/M are plotted versus M (for even values of M) for the coded channel, $E_b/N_0 = 8.0$ dB, for $N = 32$ and $N = 256$. For smaller N, the performance becomes limited by the channel at low values of M. The network capacity flattens out above $M \approx 15$, so the user capacity goes roughly as $1/M$. In the case of the coded channel, with $N = 256$, the system is not limited by the channel errors except for $M \approx 60$. For this value of N, the user capacity is roughly constant, except for very small M, and is seen to approach the value of 0.172 found by Sousa and Silvester [47], which was referred to in §4.2.3. As M increases above 60, the performance will be limited by the channel errors, and the network capacity will flatten out.

The performance of a channel load sense system with a coded channel is shown in Fig. 5.12 for a network size $M = 20$. Because this channel is better than the uncoded channel considered in the last chapter, the capacity continues to increase as K increases to 5. This is shown more clearly in Fig. 5.13, which plots the capacity C as a function of K for several values of N. Again, for low spreading factors,

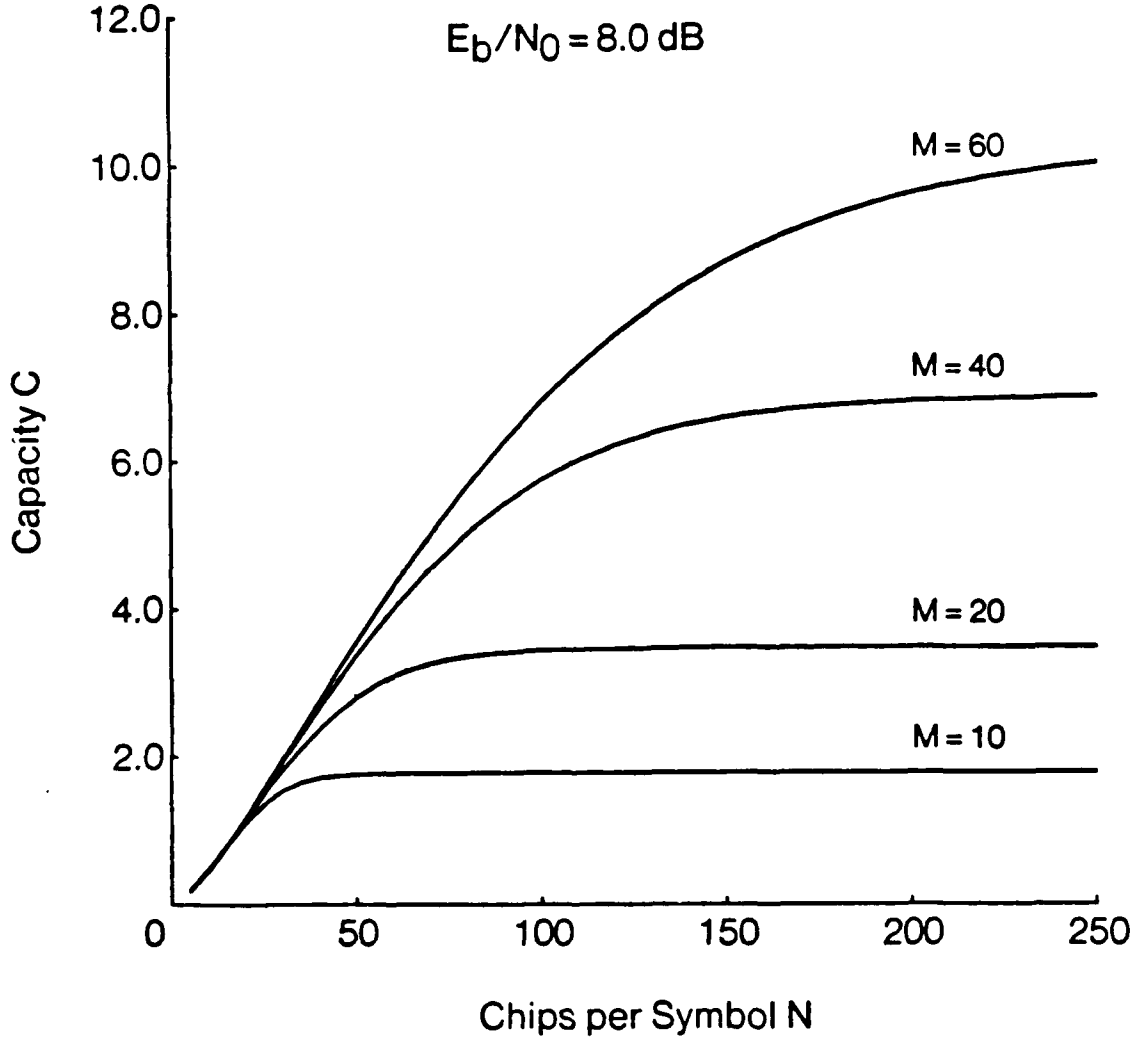


Fig. 5.10 Capacity versus spreading factor and network size, coded channel.

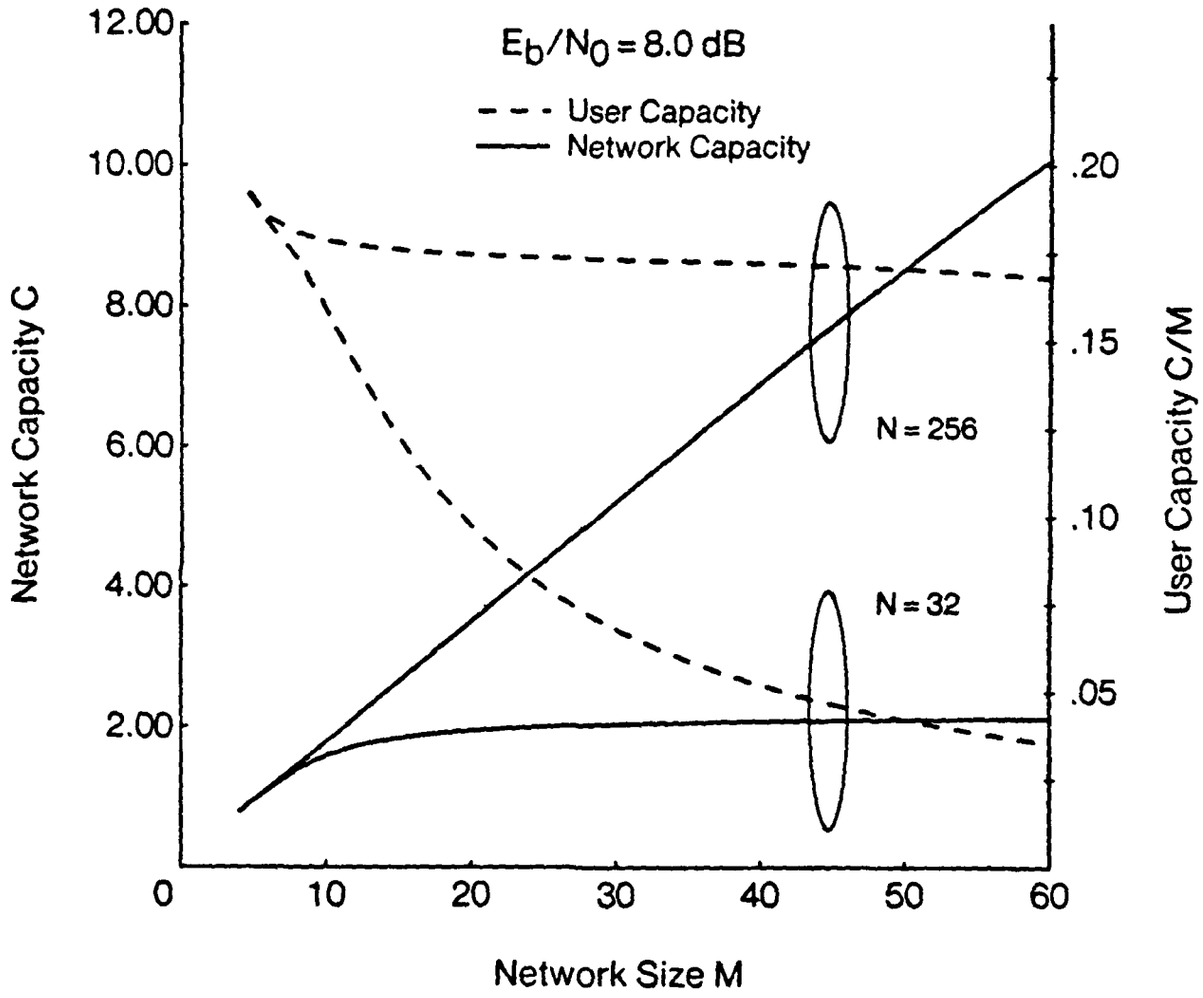


Fig. 5.11 Network capacity and user capacity versus network size, coded channel.

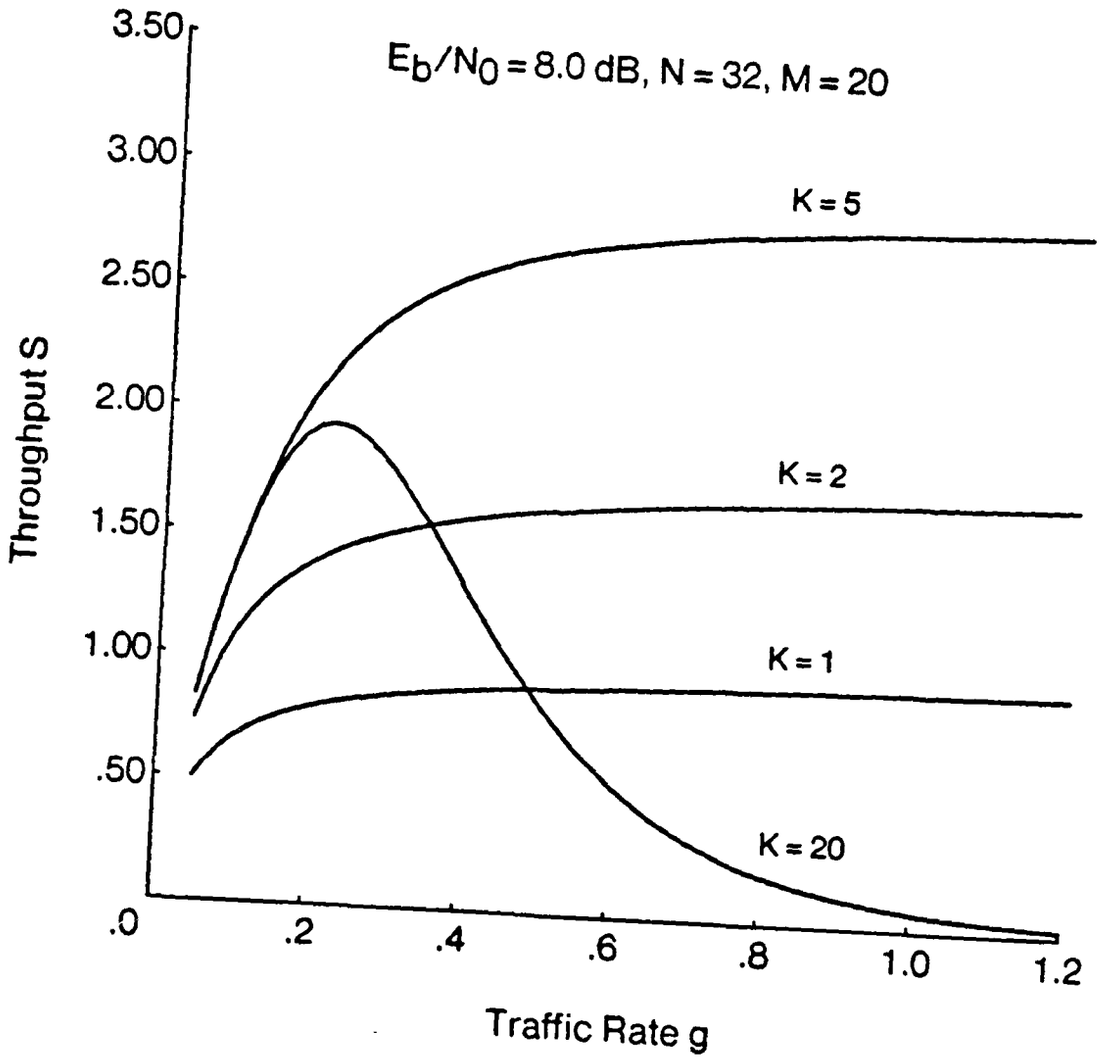


Fig. 5.12 Throughput versus traffic rate for channel load sensing, zero propagation delay, coded channel.

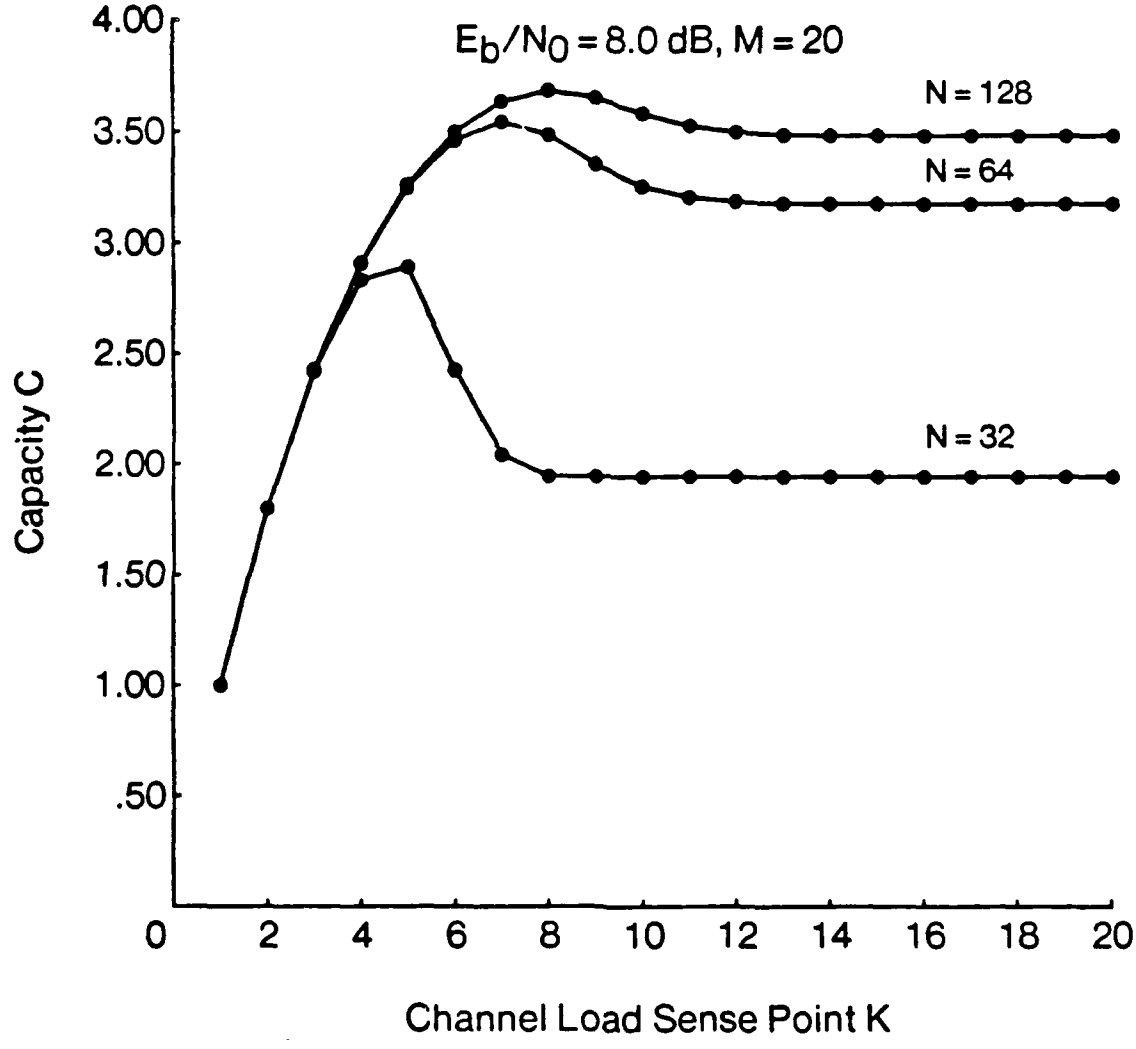


Fig. 5.13 Capacity versus channel load sense point, zero propagation delay, coded channel.

the improvement can be significant, but the improvement is only minor for larger spreading factors. Also, the improvement degrades for non-zero propagation delay, as shown in Fig. 5.14.

5.7 Summary

In this chapter, a bound on the probability that a packet is successfully decoded was found for a system with varying probability of symbol error. This bound was then approximated by considering the maximum and minimum probability of symbol error occurring over an interval of 35 bits. The approximate bound was evaluated numerically as a finite product. Because the network state changes slowly over an interval of 35 bits, especially for the finite population models, the probability of correct decoder decision can be approximated as a memoryless function of the current channel state. This allows the overall network model to be used for a system with FEC coding.

Numerical results demonstrated the improvement due to the use of FEC coding. Comparing the results of Chapter 4 for the uncoded channel with an E_b/N_0 of 10.0 dB to those presented in this chapter with a lower E_b/N_0 of 8.0 dB, for the same spreading factor W, the coded channel throughput is uniformly better than the uncoded channel throughput. Also, qualitatively, the results for coded channel approach the limiting values at lower spreading factors or received powers, and come closer to the limiting values. In other words, for the coded channels, the knees of the curves occur sooner and are sharper, and the plateaus are flatter than for the uncoded channels. As was found for the uncoded channel, the use of channel load sensing improved the capacity, especially for low spreading factors, but had little effect for the case of larger spreading factors or for moderate propagation delays.

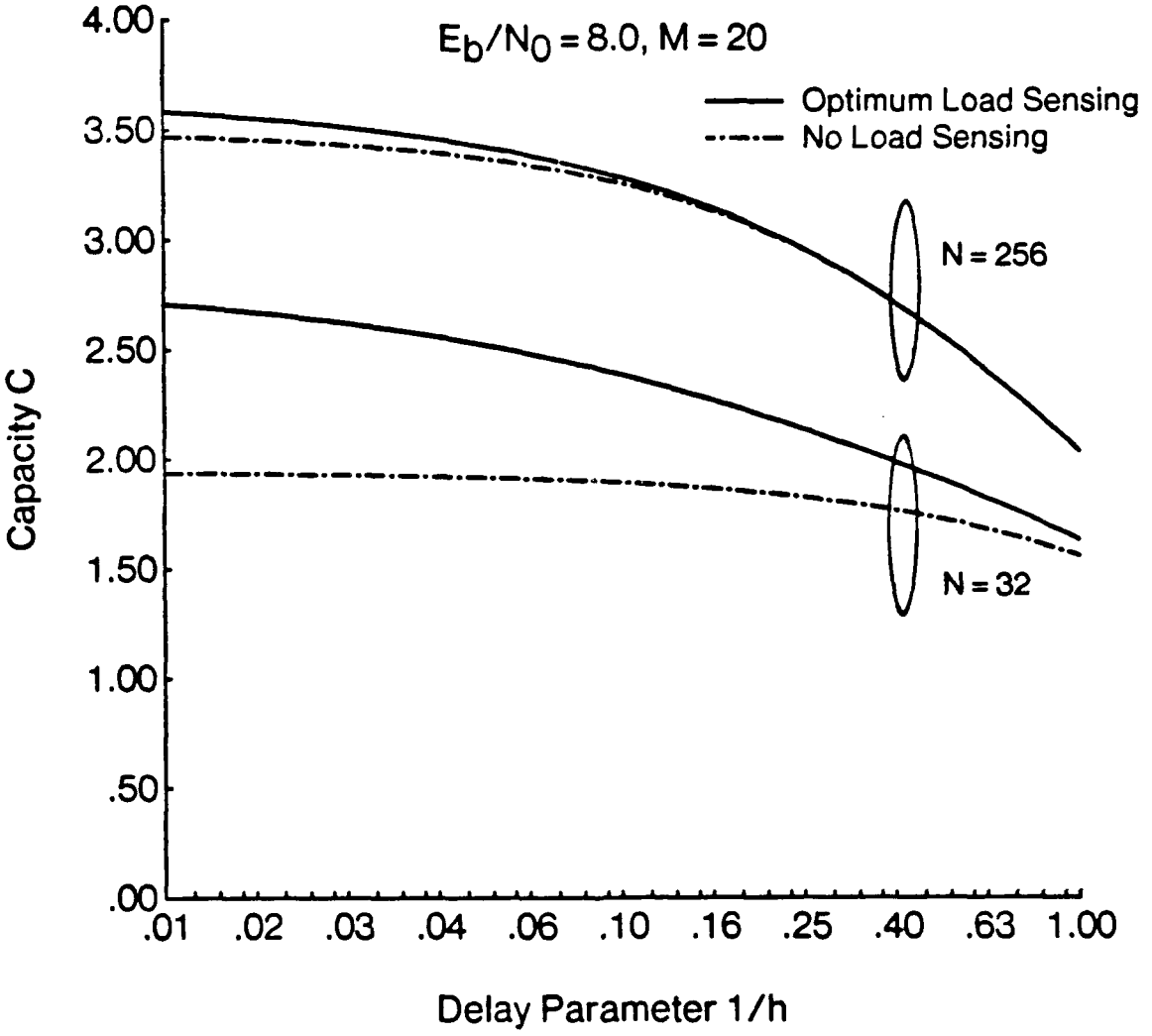


Fig. 5.14 Capacity versus propagation delay, channel load sensing, coded channel.

Because many actual networks make use of FEC coding, it is important to analyze the performance of a system with a coded channel. The approximations presented in this chapter allowed the network throughput and capacity to be found for such systems. This allowed numerical results to be found, which indicated the qualitative and quantitative differences between the coded and uncoded channels.

Chapter 6

Preamble Synchronization

Thus far, it has been assumed that all packets are acquired with probability one. In this chapter, an accurate model of the acquisition process is derived. The task of acquiring a packet consists of detecting the presence of the packet and synchronizing the local replica of the spreading code to the code used in transmission. The packets are prefaced by a preamble, which is a known sequence of chips with no data modulation. The detection and synchronization processes must be completed by the end of the preamble for acquisition to occur.

The performance of the synchronization circuit is analyzed using techniques from the study of radar systems, as the task is to accurately detect the presence or absence of a signal that is received along with noise and interference. The fundamental trade-off is between the probability of missing a packet versus the probability of erroneously indicating the presence when no packet is being received. For a given synchronization scheme, the relationship between the probability of detection and the probability of false alarm is a function of the thermal noise, the interfering signals, the received signal power, and the integration time. Therefore, the probabilities of detection and of false alarm depend upon the history of the network

state, in particular, on X, during the preamble. These probabilities in turn affect $P_{Rx}(X, R)$, the overall probability that a packet is received, and therefore affect the evolution of the network states.

The model of the system including synchronization is discussed in §6.1. The analysis of the synchronization process can be broken down into two parts. The first part is a determination of the detection and the false alarm probabilities as a function of the network state throughout the preamble. By bounding the interference level, bounds on the probabilities can be found. In §6.2, the approximate detection and false alarm probabilities are determined from the level of interference throughout the preamble. With the approximate model, the probabilities are a function of the average value of the interference over the interval of correlation. The derivation relies heavily on work done by Polydorous and Weber [36]. The second part, given in §6.3, is an analysis of the evolution of the network during the reception of the preamble. Finally, numerical results for throughput are given in §6.4.

6.1 Introduction

There are a number of variables which affect the success of the preamble synchronization process. For this reason, an exact analysis would require a greatly expanded state space, which included all the necessary variables. Furthermore, the model would no longer be Markovian, both because the preamble is of fixed length and because for some variables it is the evolution, not just an instantaneous value, which affects the synchronization. Nevertheless, by considering these variables independently for each packet, their probability distributions can be found given the network state (X, R) at the start of the packet transmission. This will allow the determination of an accurate average probability that a packet is received. A further approximation, needed to retain the Markovian nature of the network model,

is that the packet lengths are exponential, even though this length includes a fixed length preamble. This can be considered to be an approximation to a system in which the c.d.f. of the packet lengths is

$$F(\tau) = \begin{cases} 0 & \tau < T_p; \\ 1 - e^{-\mu\tau} & \tau \geq T_p, \end{cases} \tag{1}$$

where T_p is the preamble length. If $\mu T_p << 1$, then the difference between this distribution and a true exponential distribution will be negligible. The resulting number $P_{Rx}(X, R)$ cannot account for the correlation from packet to packet, nor can it account for the correlation with the event that there are no transmission errors. Even so, to a good approximation, these events are independent, so the average value is sufficient. The approximate model has been verified by means of a simulation model, described in Appendix B.

Because the network state specifically indicates neither the number of radios which are busy transmitting preambles rather than data, nor the number of radios receiving preambles, the exact meaning of the network state (X, R) can be interpreted in several ways. In this chapter, the state X is defined as being the number of radios which are transmitting either a preamble or data. In the actual network, a radio would not be counted as receiving until the end of the preamble, at which time it can be determined whether or not the packet is acquired. However, in the Markovian model, it is necessary that the number of receivers changes at the same time as the number of transmitters. Therefore, the approximation is made that a radio is counted as receiving at the start of the preamble. As a result, R is defined as the number of radios which are receiving the data portion of a packet, plus the number of radios which are receiving the preamble portion of a packet that they will eventually acquire. However, R does not include radios which are attempting to acquire a packet but do not do so successfully, nor, for a system with space-homogeneous

preamble code assignment, does it include radios which are synchronizing to the preamble of a packet that is not destined for them. These latter events will cause the destination to be busy, and could cause a packet to be missed, but they are accounted for separately. The true probability of reception is a function of events throughout the preamble. In the model, the state transition takes place at the start of the preamble, and so it depends upon events in the future. However, the approximate model assumes that each packet is received or not as a function of the average evolution of the network throughout the preamble, not as a function of the specific evolution encountered by the packet. Consequently, at the network level, the state (X, R) is sufficient to determine the transition probabilities, so the Markovian nature of the model is retained.

The simplest synchronization circuit implements a statistical decision. A random variable, resulting from integrating the received signal, is compared to a threshold. If the r.v. exceeds the threshold, the detection circuit indicates that a packet is present, and gives the rough time offset. This triggers the receiver to begin the various tasks of fine level chip synchronization and tracking, carrier synchronization and tracking, decoder initialization, etc., in order to receive the packet. While the receiver is busy executing these tasks, it is unable to receive any other packets.

When a packet is not detected, a miss is said to occur. Conversely, when a packet is indicated even though none is in fact present, a false alarm is said to occur. The performance of the detection circuit is given in terms of the probability of detection, P_d, and the probability of false alarm P_{fa}. Ideally, every packet would be detected, i.e., $P_d = 1$, and there would never be any false alarms, i.e., $P_{fa} = 0$. However, both probabilities are increasing functions of the threshold, so for a given synchronization circuit a higher P_d can only be achieved at the expense of a higher P_{fa}. If a packet is missed, it cannot be received, so the penalty of a low P_d is that

a fraction of receivable packets are unsuccessful because they are not detected. If a false alarm occurs, the receiver circuitry becomes busy for an interval of time during which it is said to be in a false lock state. The receiver cannot receive any packets arriving while it is in false lock. As P_{fa} increases, the fraction of time spent in false lock increases, causing more packets to be unsuccessful, so a high P_{fa} also ultimately leads to unsuccessful packets. Therefore, for the basic synchronization circuit, there is an optimum threshold setting which will maximize $P_{Rx}(X, R)$.

The task of synchronizing to the RF carrier of a spread spectrum signal is very difficult. It is actually easier to synchronize to the spreading code first, despread the signal, and then synchronize to the resulting narrowband carrier [49]. Consequently, the chip synchronization is non-coherent, meaning that the phase of the received signal is unknown. From texts on radar systems, such as DiFranco and Rubin [13], the optimum detector for such a signal is the I-Q (in-phase and quadrature) receiver shown in Fig. 6.1. Separate correlation circuits compare the in-phase (cosine) and quadrature (sine) components of the received signal to the known preamble code. The outputs of the correlators are squared and added, and this sum is sampled and compared to a threshold at discrete times. If there is a packet present with a time offset of within $\pm T_c$, the expected value of the signal out of the correlators will be very high, so the packet will be detected. Because the width of the main lobe of the auto-correlation function is about two chips, the detections must take place at a rate of at least once per chip if all incoming packets are to have a chance at being detected.

One physical implementation of the correlation circuit is a matched filter. With this implementation, the threshold decision can be made at a rate equal to or even greater than the chip rate. The interval of correlation is a sliding window, with a great deal of overlap between the correlation intervals of successive decisions.

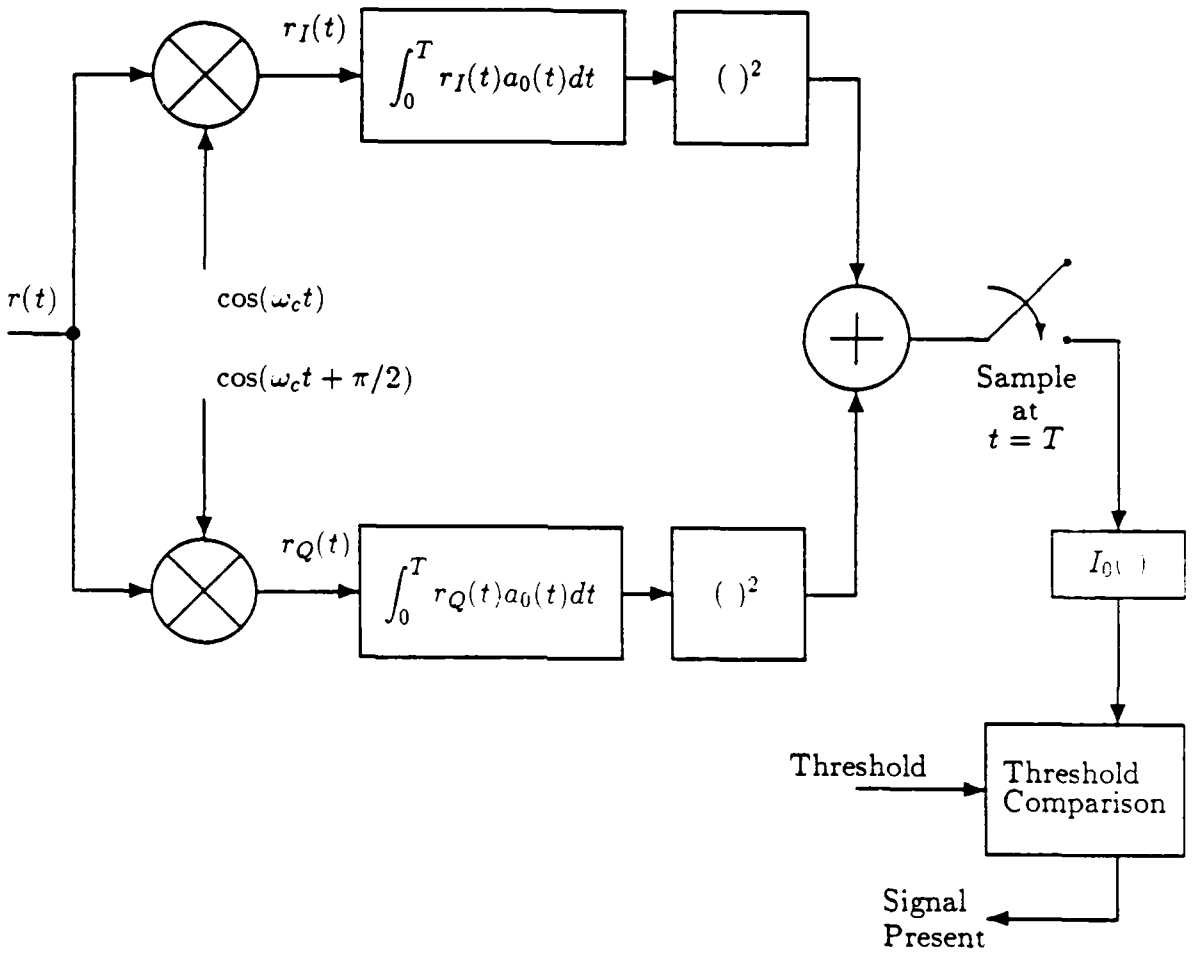


Fig. 6.1 I-Q detector.

Unfortunately, hardware constraints limit the maximum length of the correlation interval. Two devices are commonly used: surface acoustic wave (SAW) filters, and charge coupled devices (CCD's). SAW filters with a length of 1000 chips have been constructed, and a 2000 chip device is being developed [17]. CCD's are generally lower cost than SAW devices, but the maximum length is even smaller. Currently, the state of the art for CCD's is around 256 chips, but work is proceeding on a 1024 chip device [17]. Because the performance improves with longer correlation times, the limitations on the length of the matched filter can lead to a low overall $P_{Rx}(X, R)$. For example, with a spreading factor of hundreds of chips per bit, a matched filter of even 2000 chips is only several bits, which is not sufficient to adequately distinguish between true packets and false alarms.

An alternate implementation of the correlator, which does not have a restriction on maximum length, is an active correlator, shown in Fig. 6.2. In the active correlator, the incoming signal is multiplied by the local estimate of the spreading code, and the output of the multiplier is integrated over many chips. This integrated signal forms the output of the correlator. The active correlator has a serious restriction, which is that the correlation intervals cannot be overlapping, so that the rate of detection is the inverse of the integration time, which can be very low. If the integration time is equivalent to c chips, then a bank of c active correlators can be used, giving an effective detection rate of once per chip. If the approximate starting time of the packet is known, the number of active correlators required can be reduced, as offsets for which there is very little probability of a packet being present need not be considered. Nevertheless, in the unslotted ALOHA packet system, the starting times are completely random, and even for small values of c, the hardware cost of the required bank of correlators would be enormous. Thus, both the matched filter and the active correlator circuits have serious drawbacks when used by themselves as the correlation circuit.

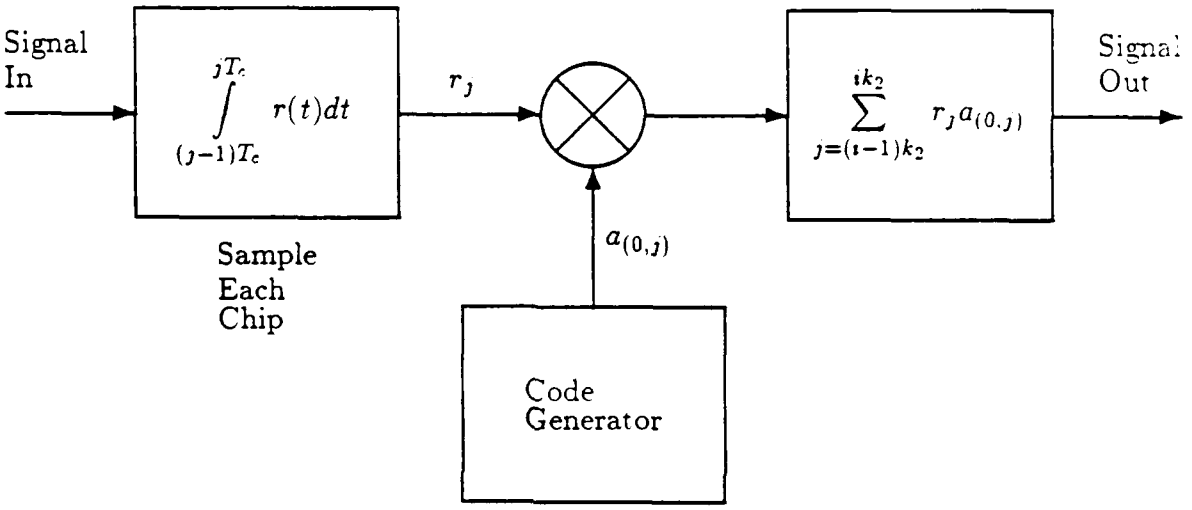


Fig. 6.2 Active correlator circuit.

Fortunately, by following a matched filter detector with an active correlator detector, a circuit with neither of the respective shortcomings can be constructed. This two stage acquisition circuit of Fig. 6.3 was proposed by Rappaport and Schilling in [44]. The preamble for such a circuit, shown in Fig. 6.4(a), is made up of two parts; the first part is the code to which the matched filter is matched, and the second part is the code which is generated by the active correlator. The detections of the matched filter, some of which may be false, indicate to the active correlator the approximate starting time of a packet. Because they will be weeded out by the active correlator (with high probability), a relatively high false alarm rate can be tolerated for the matched filter, leading to a high probability of detection. Thus, with a two stage structure, the overall performance of the active correlator can be approached without the need for a large number of active correlator circuits.

There are two ways to further improve the acquisition performance. The first method increases the overall probability of detection of the matched filter, and the second method allows a higher false alarm rate out of the matched filter to be tolerated. First, in the simplest two-stage acquisition circuit, the matched filter has only one chance at detecting the preamble. By repeating the entire preamble structure several times, as in Fig. 6.4(b), the matched filter is given more chances at detection, and therefore the probability of detecting at least one is increased. This is referred to as the multiple cycle case, and each repetition is called a cycle. For the two stage circuit, false alarms out of the matched filter have two effects. First, for a given P_{fa} of the active correlator, a higher P_{fa} for the matched filter will lead to a higher overall false alarm rate, causing more packets to be missed due to false locks. Second, every false alarm from the matched filter causes the active correlator to be busy for the entire correlation time. If the active correlator is busy when the matched filter detects a true packet, the packet will be lost. This second effect is usually the more damaging one. However, if there is more than one active

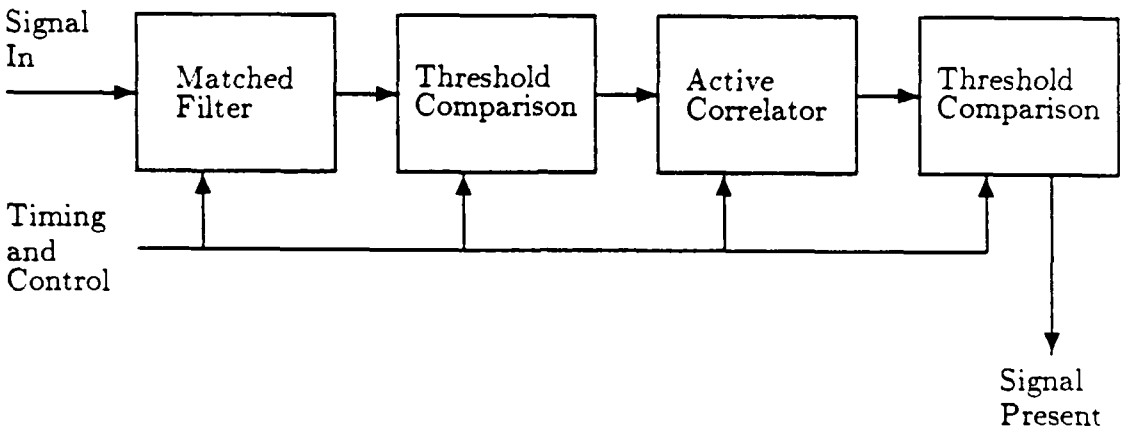


Fig. 6.3 Two stage acquisition circuit.

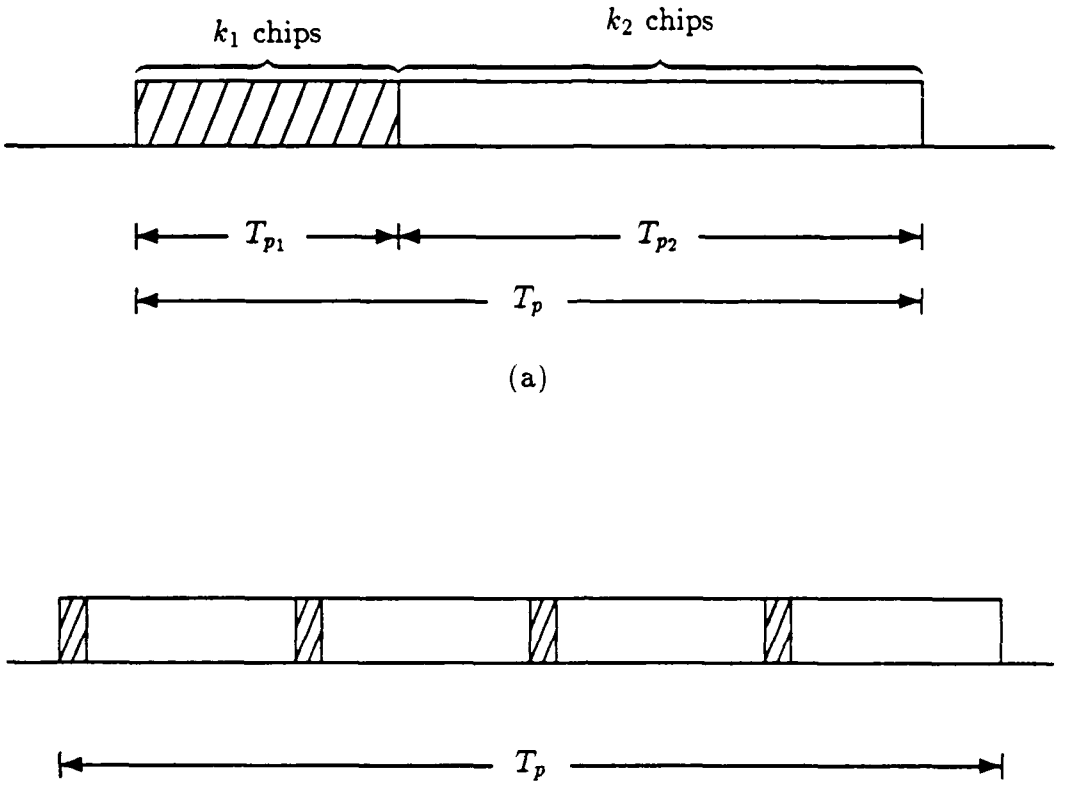


Fig. 6.4 Preamble Structure.

correlator circuit, then even if there are many false alarms in a short interval, there still might be an active correlator available for the true packet. The analysis will consider both the cases of many cycles n and of many active correlators c.

6.2 Probability of Detection and False Alarm

The performance of the synchronization circuit has been analyzed by Polydorous and Weber in [36]. They considered the detection of a single signal in the presence of Gaussian noise, and determined the effect of the auto-correlation of the spreading code. The analysis can be applied to both matched filter and active correlator I-Q detector circuits. Siess and Weber extended the analysis to a system with jamming in [48]. In this section, a similar treatment leads to close approximations to the probabilities of detection and false alarm for a multiple access system with a constant number of interfering signals throughout the preamble. It is then shown that for a system in which the interference varies, the probabilities are functions of the average level of interference.

The following notation is used. The first stage is referred to as the detection stage and the second as the verification stage. The preamble (or a cycle of the preamble in the multiple-cycle case) is divided into two parts. The length of the first part, k_1 chips, is equal to the length of the matched filter. The length of the second part, k_2 chips, is equal to the length of the active correlator. The thresholds are denoted by l_1 and l_2. The preamble has a duration of T_p, which is the sum of the duration of the first part, $T_{p_1} = k_1 T_c$ and duration of the second part, $T_{p_2} = k_2 T_c$, with a chip duration of T_c. The probabilities of detection and of false alarm for the first and second stage are denoted by $P_{d_1}(I)$, $P_{fa_1}(I)$, $P_{d_2}(I)$, and $P_{fa_2}(I)$, where I is the constant level of interference occurring during the preamble under

consideration. H_1 denotes the event that a preamble (on the same code) with a time offset τ_1 less than $\pm T_c$ is present, and H_0 the complementary event. The analysis treats the performance of the matched filter, yielding $P_{d_1}(I)$ and $P_{fa_1}(I)$. Substituting k_2 and l_2 for k_1 and l_1 will give the values of $P_{fa_2}(I)$ and $P_{d_2}(I)$.

6.2.1 Constant Interference Level

As in Chapter 2, the incoming signal $r(t)$ is

$$r(t) = \sum_{k=0}^{I} \sqrt{2S}\tilde{a}_k(t-\tau_k)\cos(\omega_c t + \phi_k) + \mathcal{N}, \tag{6.2}$$

where $\tilde{a}_k(t)$ is the transmitted chip sequence, equal to the spreading code times the data sequence, and the desired signal is now indexed by $k = 0$. After multiplying $r(t)$ by $\sqrt{2}\cos(\omega_c t)a_0(t)$ or $\sqrt{2}\sin(\omega_c t)a_0(t)$ and integrating, the in-phase and quadrature signals (neglecting double frequency terms) are

$$\begin{aligned} \mathcal{Z}_I &= T_{p_1}\sqrt{S}\sum_{k=0}^{I} A_k\cos(\phi_k) + \mathcal{N}_I; \\ \mathcal{Z}_Q &= T_{p_1}\sqrt{S}\sum_{k=0}^{I} A_k\sin(\phi_k) + \mathcal{N}_Q, \end{aligned} \tag{6.3}$$

where $\mathcal{N}_I$ and $\mathcal{N}_Q$ are independent zero-mean Gaussian r.v.'s, each with variance $N_0 T_{p_1}/2$, and A_k is the aperiodic partial cross-correlation of $\{a_{(0,j)}\}$ and $\{\tilde{a}_{(k,j)}\}$,

$$A_k = \frac{1}{k_1}\sum_{j=1}^{k_1}(1-\delta_k)\tilde{a}_{(k,j')}a_{(0,j)} + \delta_k\tilde{a}_{(k,j'-1)}a_{(0,j)}; \qquad j - j' = J_k. \tag{6.4}$$

With random coding, for $k \neq 0$, A_k is the sum of a number of Bernoulli random variables. Thus, for $k \neq 0$, the mean of A_k is zero and the variance is $(1-\delta_k)^2 + \delta_k^2$, where δ_k is the fractional part of τ_k as in Chapter 2. Given the event H_0, A_0 is also

zero mean, with variance $(1-\delta_0)^2+\delta_0^2$. Given the event H_1, that is, a preamble is present with offset $|\tau_0|<1$, the mean of A_0 is $(1-|\tau_0|)$, and the variance is τ_0^2.

The components of $\mathcal{Z}_I$ and $\mathcal{Z}_Q$ excluding thermal noise can be expressed in polar coordinates as

$$\mathcal{Z}_I-\mathcal{N}_I=y\cos(\theta) \qquad \text{and} \qquad \mathcal{Z}_Q-\mathcal{N}_Q=y\sin(\theta). \tag{6.5}$$

The value of θ does not affect the detection and false alarm probabilities. The value of y is found from

$$\begin{aligned} y^2 &= ST_{p_1}^2\left(\left(\sum_{k=0}^{I}A_k\cos(\phi_k)\right)^2+\left(\sum_{k=0}^{I}A_k\sin(\phi_k)\right)^2\right) \\ &= ST_{p_1}^2\left(\sum_{k=0}^{I}(A_k)^2+\sum_{k=0}^{I}\sum_{\substack{l=0\\l\neq k}}^{I}A_kA_l(\cos(\phi_k)\cos(\phi_l)+\sin(\phi_k)\sin(\phi_l))\right) \\ &= ST_{p_1}^2\left(\sum_{k=0}^{I}(A_k)^2+\sum_{k=0}^{I}\sum_{\substack{l=0\\l\neq k}}^{I}A_kA_l\cos(\phi_k-\phi_l)\right). \end{aligned} \tag{6.6}$$

For a given set of A_k, the expected value over all ϕ_k of y^2 is simply

$$E(y^2)=(T_{p_1})^2S\sum_{k=0}^{I}(A_k)^2. \tag{6.7}$$

In order to lead to tractable results, the approximation is made that the phase offsets ϕ_k are identical for all k. Because this approximation does not affect the mean or variance of y, it is believed that it will not have a great impact on the results. As a result of the approximation, $\cos(\phi_k-\phi_l)=1$ for all k and l, and

$$y^2=(T_{p_1})^2S\left(\sum_{k=0}^{I}A_k\right)^2. \tag{6.8}$$

so y is the sum of $2(I+1)$ (for H_0) or $2I+1$ (for H_1) binomial random variables, with different variances. Using the approximation that y is Gaussian distributed (valid for large k_1 and I), allows the analysis of Polydorous and Weber to be followed almost verbatim, as was done for the case of a CW jammer, studied by Siess and Weber [48]. The mean and variance of y for the hypotheses H_i are

$$m_i = \begin{cases} 0 & i = 0; \\ T_{p_1}\sqrt{S}(1 - |\tau_0|) & i = 1; \end{cases} \tag{6.9}$$

$$\sigma_i^2 = \begin{cases} k_1(T_c)^2 S \sum_{k=0}^{I} G_0(\delta_k) & i = 0; \\ k_1(T_c)^2 S \left(G_1(\tau_0) + \sum_{k=1}^{I} G_0(\delta_k) \right) & i = 1, \end{cases} \tag{6.10}$$

where

$$G_0(\delta) = 1 - 2\delta + 2\delta^2 \qquad \text{and} \qquad G_1(\tau) = \tau^2. \tag{6.11}$$

Again, it is not possible to integrate over all offsets δ_k. The worst case occurs when $\delta_k = 0$ for all k. The best case is $\delta_k = 1/2$. Alternatively, $\sum_{k=1}^{I} G_0(\delta_k)$ can be approximated by the average value, $(2/3)I$. For the hypothesis H_0, $G_0(\delta_0)$ is set equal to the value used for the other interfering transmissions, regardless of the actual value of τ_0 used to find P_{d_1}. Following the approach of Polydorous and Weber [36], the effective SNR is

$$SNR_e^i = \begin{cases} E_c/N_0(\eta(I+1)) & i = 0; \\ E_c/N_0(\eta I + \tau_0^2) & i = 1, \end{cases} \tag{6.12}$$

where η is 1, 2/3 or 1/2. The probability of false alarm is

$$P_{fa_1} = \kappa \int_{l_1^*}^{\infty} \exp(-x) I_0(a_* x) dx, \tag{6.13}$$

$$\kappa = \frac{\sqrt{1+2SNR_e{}^0}}{1+SNR_e{}^0}; \qquad l_1^* = \frac{1+SNR_e{}^0}{1+2SNR_e{}^0}\frac{l_1}{2}; \qquad a_* = \frac{SNR_e{}^0}{1+SNR_e{}^0}. \tag{6.14}$$

The numerical value of P_{fa_1} is calculated from the series expansion

$$P_{fa_1}(I) = \sum_{k=0}^{\infty} f_k \sum_{m=0}^{2k} \frac{(l_1^*)^m}{m!}\exp(-l_1^*); \tag{6.15}$$

$$f_0 = \kappa; \quad f_k = \frac{k-1/2}{k} a_*{}^2 f_{k-1}. \tag{6.16}$$

The probability of detection is calculated from the series expansion

$$P_{d_1}(I) = \exp\left\{-\frac{k_1(1-|\tau_0|)^2 E_c/N_0}{1+2SNR_e{}^1}\right\}\frac{1}{\sqrt{1+2SNR_e{}^1}}\sum_{k=0}^{\infty} F_k G_k, \tag{6.17}$$

where G_k is found from the recursion equation

$$G_{k+1} = (i_1/2)^{k+1}e^{-l_1/2} + (k+1)G_k; \qquad G_0 = e^{-l_1/2}, \tag{6.18}$$

and F_k is found from

$$F_0 = 1; \qquad E_1 = m_*/2; \tag{6.19}$$

$$F_{k+1} = \frac{(k+1/2)}{(k+1)^2}\sigma_*^2 F_k + m_* E_{k+1}; \tag{6.20}$$

$$E_{k+1} = \frac{1}{(k+1)^2}(m_* F_k/2 + \sigma_*^2 k E_k); \tag{6.21}$$

$$m_* = \frac{\sqrt{2k_1(1-|\tau_0|)^2 E_c/N_0}}{1+2SNR_e{}^1}; \tag{6.22}$$

$$\sigma_*^2 = \frac{2SNR_e{}^1}{1+2SNR_e{}^1}. \tag{6.23}$$

The output of the matched filter is sampled at intervals of ξT_c. This will lead to a worst case time offset of $|\tau_0| = \xi/2$, for both the matched filter and the active correlator. Also, for this worst case offset, there are two chances at detection [35], so the overall P_d^{ov} is

$$P_d^{ov} = P_d + (1 - P_d)P_d. \tag{6.24}$$

6.2.2 Varying Interference Level

Using the approximations of the previous analysis, P_{d_1} and P_{d_2} can be evaluated even if the interference varies during the detection. Several new variables are needed. Let l be the number of times that the number of interfering transmitters changes. The times at which the interference changes are denoted by $t_1, t_2, \ldots, t_l$, and the start of the tagged preamble is assumed to be at time $t = 0$. Let I^* be the total number of packets that overlap some part of the interval $[0, T_{p_1}]$. Instead of A_k, the variables B_k are needed, defined as

$$B_k = \int_0^{T_{p_1}} a_0(t)\tilde{a}_k(t - \tau_k)dt. \tag{6.25}$$

If the kth packet overlaps the entire interval $[0, T_{p_1}]$ then $A_k = B_k/T_{p_1}$. It is again assumed that ϕ_k is the same for all k, so the r.v. y is equal to

$$y = S\sum_{k=0}^{I^*} B_k. \tag{6.26}$$

B_0 is simply $T_{p_1}A_0$, so it can be approximated as a Gaussian r.v., as done previously. If the kth interfering packet overlaps the entire interval, then B_k can be approximated as a zero mean Gaussian r.v., with variance $(T_{p_1})^2\eta/k_1 = k_1(T_c)^2\eta$. If the kth packet does not overlap the entire interval, the Gaussian approximation

might not be valid, as the binomial r.v. can be the sum over very few chips. However, when the overlap is this small, the interference from the kth packet will have very little effect on the performance, so the inaccuracy of the approximation will not matter. Thus, to a good approximation, the variable B_k will be Gaussian, with zero mean, and variance $N_k(T_c)^2\eta$, where N_k is the number of chips in $[0, T_{p_1}]$ that the kth packet overlaps. Consequently, the sum y is Gaussian, with a mean given by Eqn. 6.9 and a variance of

$$\sigma_i^2 = \begin{cases} T_c S \int_0^{T_{p_1}} \eta(I(t)+1)dt, & i = 0; \\ T_c S \left(T_{p_1} G_1(\tau_0) + \int_0^{T_{p_1}} \eta(I(t))dt \right), & i = 1, \end{cases} \tag{6.27}$$

where $I(t)$ is the instantaneous level of interference at time t. However, this is precisely the expression of Eqn. 6.10, where the constant value I is replaced by the average over the interval $[0, T_{p_1}]$ of the interference $I(t)$.

6.3 Analysis of the Probability of Reception

In this section, the average probability that a packet is received is determined for the single cycle, single active correlator, two stage acquisition circuit described in §6.1. Specifically, the analysis yields $P_{Rx}(X, R)$, the probability that a packet is received given that the network state at time $t = 0^-$ is (X, R), and the source of the packet was idle and began transmission to a randomly chosen destination at time $t = 0$. The packet under consideration is called the tagged packet, and its preamble, the tagged preamble. The tagged preamble extends from time $t = 0$ to $t = T_p$, with the first part ending at T_{p_1}.

6.3.1 Events Constituting Reception

For a packet to be received, a number of events must take place successfully. If the synchronization circuitry is not used at all following the detection of a packet, and the time to switch the PRU from a receiving state to a state of searching for new packets is negligible compared to T_p, then it may be possible for the destination to acquire a packet whose preamble begins while that destination is receiving another packet, as long as the reception ends before the end of the preamble. However, for many systems, this will not be the case, so in this analysis it is assumed that if a radio is busy receiving at the start of a preamble, then that preamble cannot be acquired, even if the radio becomes idle before the end of the preamble. Also, since the radios cannot receive and transmit simultaneously, a packet will not be received if the destination transmits at any time during the preamble. Thus, the first condition for reception to occur is that the destination must be idle throughout the preamble. Next, the matched filter must detect the first part of the preamble. At the end of the first part, the active correlator must be idle. Also, the active correlator must verify the preamble. Finally, the destination receiver must not be in false lock. These events are described in more detail below.

The protocol allows radios to transmit while attempting to acquire a packet. Therefore, P_{DI}, the probability that the destination is idle (given a state at time $t = 0^-$ of (X, R) and that the source was idle and begins transmission), must take into account the state of the destination throughout the entire preamble. Because of the definition of the state R, if a receiver is not counted as one of the R at time $t = 0^-$, then no packet that will be acquired has yet begun transmission. In other words, if the destination begins receiving a packet in the interval $(0, T_p)$, that packet's preamble will end in the interval $(T_p, 2T_p)$. Because the new preamble ends after the tagged preamble, it cannot cause the tagged packet not to be acquired due

to the destination being busy. Of course, the new preamble will affect the acquisition probability by increasing the interference level during $(0, T_p)$, but this effect will be considered separately. Therefore, to find P_{DI}, it is necessary to determine whether the destination was transmitting at any time during the preamble, but it is only necessary to know whether the destination was included in the R radios receiving at the beginning of the preamble. So, P_{DI} can be found as the probability that the destination is not transmitting or receiving at time $t = 0^-$, times the conditional probability that the destination does not begin transmitting during $(0, T_p)$, given that it was idle at $t = 0^-$.

Given that the destination is idle throughout the preamble, the matched filter has a chance at detecting the first part of the preamble. The performance $P_{d_1}(I)$ was found for a constant level of interference in §6.2. It will be shown in §6.3.3.2 that the probability of detection for an arbitrary network evolution during $(0, T_p)$ can be bounded by $P_{d_1}(I)$ for appropriate values of I.

If the matched filter successfully detects the preamble, an active correlator will begin verification, if one is available. The probability that there is an active correlator available, given that the destination is idle, given the evolution of the interference over the infinite past, and given the number of overlapping preambles on the same code, is denoted by P_{NV}. This probability is independent of the event that the preamble was detected, when conditioned on the synchronization parameters l_1 and k_1. There are two processes affecting P_{NV}. The first is the false detections of the matched filter, which lead to a correlator verifying a false signal. The second is the true detections of the overlapping preambles. For the simple two-stage preamble with one cycle and one correlator, the probability that the correlator is not busy verifying a false detection, conditioned on the events outlined earlier, is denoted by P_{NVF}. The probability that the correlator is not verifying an overlapping preamble,

again with the same conditions, is denoted by P_{NVT}.

Conditioned on all of the earlier events, i.e., the destination is idle, the matched filter detects the preamble, and an active correlator is available, the probability of correct verification is only a function of the interference, the chip offset, and the parameters E_c/N_0, k_2, and l_2. As was the case for the matched filter, this can be bounded by considering $P_{d_2}(I)$ for a constant I, using the appropriate value of I.

Finally, if the active correlator correctly verifies the preamble, there is still one more hurdle, which is false lock. Because the synchronization process is imperfect, there will be times when the circuitry indicates the presence of a packet when none is there. This will cause the receiver to be tied up processing a non-existent packet. If a true packet arrives during this time, it will be lost because the receiver is busy. There are several possible mechanisms that will detect a false lock condition. First, in order to continue despreading the packet, the PRU must have a code tracking loop. This loop will include lock detection circuitry which will indicate when the tracking has lost lock on a signal. If a false lock occurs, the tracking loop will quickly indicate that it has lost lock of the signal. A second mechanism is the packet header, which contains data such as the source and destination addresses. Even if the tracking loop does not indicate false lock, the header will be random bits, so the packet will be rejected. Ideally, the PRU would immediately return to an idle state when loss of lock occurs, so the false lock condition will quickly be corrected. However, there is likely to be overhead involved in initiating and terminating the reception of a packet. Therefore, the duration of the false lock is highly system dependent. For simplicity, the penalty for a false lock will be modeled as a fixed time interval during which the PRU is inactive.

6.3.2 Auxiliary Variables

For an exact analysis, the state description must be rich enough to allow the

various probabilities defined in §6.3.1 to be determined directly. However, a model with such a state would be too complex to lead to tractable results, and also would be non-Markovian. Rather than including these variables, called auxiliary variables, in the state description, they are found probabilistically from the simple network state (X, R). In this section, these auxiliary variables are introduced and the probability distribution functions are evaluated. These will be used in the next section to determine the probabilities affecting the overall probability of reception $P_{Rx}(X, R)$.

By making the approximation that the preamble length is a negligible fraction of the total packet length, the exponential distribution can still be used for the packet lengths, and accordingly, the Markovian network model can be retained. This approximation is used at the network level to determine steady state probabilities and transition probabilities. Once the approximate network evolution has been determined, the analysis of the synchronization process can specifically consider the existence of a fixed length preamble. Thus, the packet lengths are considered to be exponentially distributed whenever the network transitions are examined. When the synchronization process is analyzed as a function of these transitions, the fixed length preamble is accounted for. Of course, this leads to an inconsistency between the model of the synchronization process and the overall system model, but this will be inconsequential as long as the preamble is only a small fraction of the mean packet length. As will be discussed when presenting the results, this condition is true for almost all of the cases studied. The final validation comes from the simulation results, which have shown very good agreement with the approximate analytical results.

6.3.2.1 Definitions

First, in order to find P_{NV}, it is necessary to know the probability of a detection which might tie up a correlator at time T_{p_1}. As seen in Fig. 6.5, the only detections

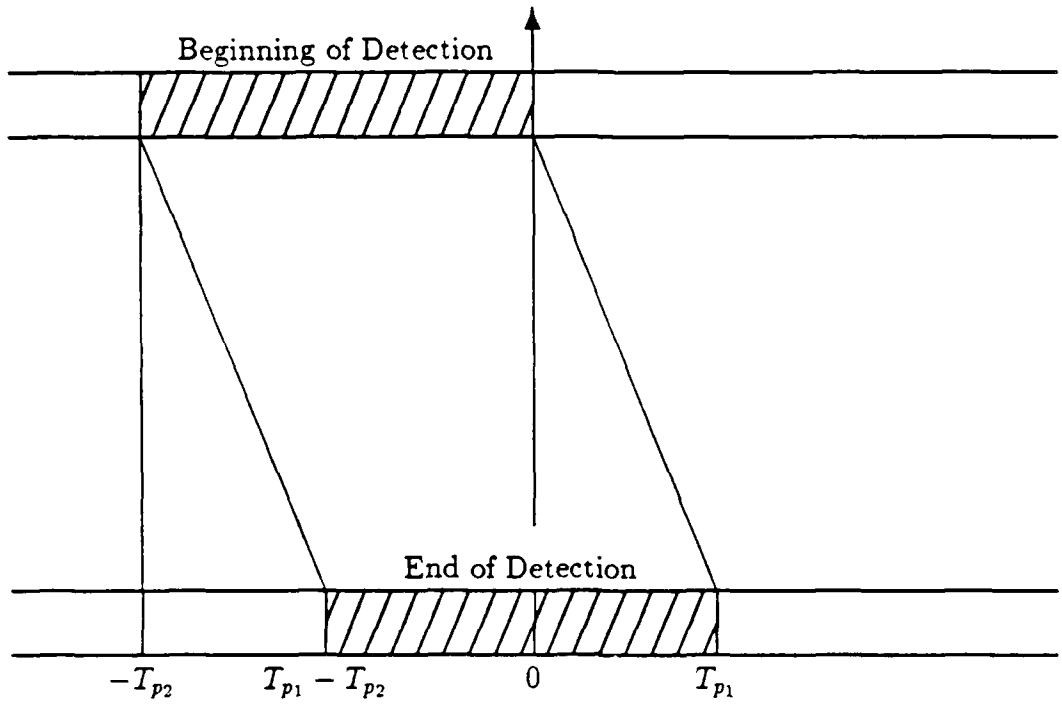


Fig. 6.5 Detections which might inhibit reception.

which might cause the correlator to be busy at time T_{p_1} are those occurring in the interval $(T_{p_1} - T_{p_2}, T_{p_1})$. These detections are a function of the interference throughout the interval $(-T_{p_2}, T_{p_1})$, which is approximated by the constant value $\hat{I}_0$. In addition, a true verification can occur only if there is another packet present. The preambles which might be detected in the interval $(T_{p_1} - T_{p_2}, T_{p_1})$ correspond to packets which begin in the interval $(-T_{p_2}, 0)$. Furthermore, if receiver directed preamble codes are used, only those preambles using the same code can be detected by the destination. The number of packets on the same code starting in this interval is denoted by V. The probabilities of detection and verification are functions of the average interference encountered during the first and second parts of the preamble. As discussed in §6.3.3.2, if P_{d_1} and P_{d_2} are greater than 1/2, they can be bounded by considering the maximum and minimum values of the interference occurring throughout the entire preamble, denoted by I^+ and I^-. The approximate values of P_{d_1} and P_{d_2} are found by considering an approximation to the to the average interference, denoted by $\hat{I}$. In order to determine I^+ and I^-, a number of other variables must be introduced. The auxiliary variables that are considered in the analysis are listed below.

V: The number of packets using the same preamble code which begin transmission in the interval $(-T_{p_2}, 0)$.

$\hat{I}_0$: An approximation to the level of interference occurring throughout the interval $(-T_{p_2}, T_{p_1})$.

I^+: The maximum level of interference in the interval $(0, T_p)$.

I^-: The minimum level of interference in the interval $(0, T_p)$.

$\hat{I}$: An approximation to the level of interference occurring throughout the interval $(0, T_p)$.

Y_0: The number of packets which begin transmission in the interval $(-T_p, 0)$.

$\tilde{X}$: The number of transmissions which end in the interval $(0, T_p)$.

$\tilde{R}$: The number of receptions which end in the interval $(0, T_p)$.

J: The number of radios which are idle at some time during $(0, T_p)$.

Y_1: The number of packets which begin transmission in the interval $(0, T_p)$.

Y_1^+: An upper bound on Y_1.

Y_1^-: A lower bound on Y_1.

Y^+: The maximum number of overlapping preambles (on any code) in the interval $(0, T_p)$.

Y^-: The minimum number of overlapping preambles (on any code) in the interval $(0, T_p)$.

The variables $\tilde{X}$, $\tilde{R}$, and J are introduced in order to determine Y_1^- and Y_1^+, which in turn are used along with Y_0 in order to find Y^+ and Y^-. All of these lead to the bound on the interference I throughout the preamble.

6.3.2.2 Probability Distribution Functions

The probability distribution functions of the auxiliary variables are found by analyzing the network evolution preceding and during the preamble of the tagged packet. This evolution is determined from the Markovian network model, which assumes exponentially distributed packet lengths. First, it is shown that if the starting times of the X transmissions in progress at $t = 0$ are assumed to be independent then they are exponentially distributed, with mean $1/\mu$. Second, from the Markovian model, the ending times of these transmissions are also independent and exponentially distributed, mean $1/\mu$. Third, from the definition of the state R, discussed in §6.1, it is known that any radio which is idle at time $t = 0$ will

not synchronize to a packet before time $t = T_p$. Therefore, any radio counted as idle at time $t = 0$ is free to transmit at any scheduling point in the interval $(0, T_p)$. Also, the scheduling points are drawn from a Poisson process, so the distribution of the time until an idle radio begins transmission is identical to the exponential distribution for $0 \leq t \leq T_p$. From these three distributions, the auxiliary variables can be determined.

First, the distribution of the starting times of the X packets must be determined. For the actual network, these starting times may not be independent. Also, the primary Markov chain is non-reversible, so it does not immediately follow that the starting times are independent, even for the Markovian model. Nevertheless, the assumption is made that these times are independent. The distribution of the starting times of these packets can be found by considering the random process of X total packets which are exponentially distributed. The process continues for all time, so whenever a packet finishes, another begins to take its place. The elapsed times of the ongoing packets at time $t = 0$ in the primary Markov chain have the same distribution as the elapsed times of the X packets in this continuing process at a randomly chosen point in time. The time since the start of the newest packet is exponentially distributed, mean $1/\mu X$, and this is equally likely to be any of the X packets. The time between starting times preceding the most recent are also exponentially distributed, mean $1/\mu X$. Therefore, the time since the start of any of the packets in the continuing process, and therefore in the primary Markov chain as well, is exponentially distributed, with mean $1/\mu$.

Now, switching from the network model to the detailed synchronization model, the number of overlapping preambles at time 0 is easily determined. The probability that a given packet begins transmission in some interval is found from the c.d.f. of the exponential distribution. Y_0, the number of overlapping preambles (on any

code) at time $t = 0$, is equal to the number of transmissions which begin in the interval $(-T_p, 0)$. The probability that a given transmission begins in this interval is $p_0 = 1 - e^{-\mu T_p}$. Because the transmission times are independent, this leads to a binomial distribution on the number of overlapping preambles. Thus, the distribution of Y_0 is

$$\Pr(Y_0 = y_0) = \binom{X}{y_0} p_0^{y_0} q_0^{X-y_0}, \tag{6.28}$$

where $q_0 = 1 - p_0$.

The probability distribution of V is found in a similar manner, with the additional consideration for receiver directed codes that the same preamble code is used, which has a probability of $1/(M-1)$. Defining p_1 as

$$p_1 = \begin{cases} 1 - e^{-\mu T_{p_2}} & \text{Space homogeneous;} \\ \dfrac{1 - e^{-\mu T_{p_2}}}{M-1} & \text{Receiver directed.} \end{cases} \tag{6.29}$$

gives

$$\Pr(V = v) = \binom{X}{v} p_1^v q_1^{X-v}; \qquad q_1 = 1 - p_1. \tag{6.30}$$

However, V is not independent of Y_0, so the conditional distribution of V given Y_0 must also be determined. For the more approximate analysis of $P_{Rx}(X, R)$, Y_0 is not needed, in which case the distribution of V is found directly from Eqn. 6.30. From the properties of the exponential distribution, given that a packet begins in $(-T_p, 0)$, the starting time is uniformly distributed in this interval. Thus, the probability that one of the preambles counted in Y_0 begins in the smaller interval $(-T_{p_2}, 0)$ and is on the same code is equal to

$$p_2 = \begin{cases} T_{p_2}/T_p & \text{Space homogeneous;} \\ \dfrac{T_{p_2}/T_p}{M-1} & \text{Receiver directed.} \end{cases} \tag{6.31}$$

V is binomially distributed given Y_0,

$$\Pr(V = v) = \binom{Y_0}{v} p_2^v q_2^{Y_0 - v}; \qquad q_2 = 1 - p_2. \tag{6.32}$$

The number of packets completing transmission by the end of the preamble, $\tilde{X}$, can also be found by considering the c.d.f. of the exponential distribution.

$$\Pr(\tilde{X} = x) = \binom{X}{x} p_0^x q_0^{X-x}, \tag{6.33}$$

where p_0 and q_0 are as defined earlier. Conditioned on X, R, and $\tilde{X}$, the next step is to find the distribution of $\tilde{R}$, the number of radios completing reception by time T_p. There are X transmissions at time $t = 0$, R of which are also being received. $\tilde{X}$ of these are chosen randomly from the X, and $\tilde{R}$ is the intersection of $\tilde{X}$ and R. This is equivalent to the combinatorial problem of chosing $\tilde{X}$ balls from an urn, without replacement, where the urn originally contains R red balls and $X - R$ black balls. From Vol. 1 of Feller [15], $\tilde{R}$ is hypergeometrically distributed,

$$\Pr(\tilde{R} = r) = \frac{\binom{R}{r}\binom{X-R}{\tilde{X}-r}}{\binom{X}{\tilde{X}}}. \tag{6.34}$$

If the preamble is very short, the probability of any packets completing transmission during the interval $(0, T_p)$ will be negligible. Therefore, a valid approximation for short preambles is that both $\tilde{X}$ and $\tilde{R}$ are identically zero.

The total number of radios which might begin transmission at some time in the interval $(0, T_p)$, denoted by J, includes those radios that are idle at time 0, and also includes those which become idle at time $t < T_p$. The size of the former group is $M-X-R-1$, and of the latter group, $\tilde{X}+\tilde{R}$, so $J = M-X-R-1+\tilde{X}+\tilde{R}$. The latter group of radios have a smaller interval in which to begin a transmission, so are less

likely to add to the interference. The highest probability of new transmissions will occur when all of the $\tilde{X}$ packets that finish do so at time $t = 0^+$, allowing the radios the entire interval to begin a transmission. With this bound, the probability that one of the J radios begins transmission is $p_3 = 1 - e^{-\lambda T_p}$, regardless of whether that radio was counted in $\tilde{X}$ or $\tilde{R}$ or not. Furthermore, for this worst case, the starting times of packets which do begin transmission in this interval are independent and uniform. The resulting bound on the number of packets starting in $(0, T_p)$, Y_1^+, has a binomial distribution,

$$\Pr(Y_1^+ = y_1) = \binom{J}{y_1} p_3^{y_1} q_3^{J-y_1}; \qquad q_3 = 1 - p_3. \tag{6.35}$$

Similarly, a lower bound on the number of packets starting transmission in this interval is found by considering the other extreme, namely, that any radio which becomes idle after $t = 0$ does so at time $t = T_p$. In this case, $J^- = M - X - R - 1$, and the bound is

$$\Pr(Y_1^- = y_1) = \binom{J^-}{y_1} p_3^{y_1} q_3^{J-y_1}. \tag{6.36}$$

Finally, given Y_0 and the bounds on Y_1, the distributions of Y^+ and Y^-, the minimum and maximum number of overlapping preambles (on any code) during $(0, T_p)$, can be determined by using a result of Pursley [41]. The conditions required by his analysis are that the number of preambles being transmitted at the beginning and end of a fixed length interval are known, and that the ending times of the original preambles and the starting times of the new preambles are all independent and uniformly distributed over the interval. These conditions are satisfied by the original preambles Y_0. The true starting times of new preambles will not be uniform over the interval, since some of the new transmissions may be from radios which are busy at $t = 0$, and are thus more likely to occur later in the interval.

Nevertheless, the bounds Y_1^+ and Y_1^- lead to uniform distributions, as discussed previously. Therefore, as given by Pursley, the cumulative distribution of Y^+ is found from

$$\Pr(Y^+ \geq y) = \begin{cases} 1 & y \leq \max(Y_0, Y_1^+); \\ \dfrac{\binom{Y_0+Y_1^+}{y}}{\binom{Y_0+Y_1^+}{Y_0}} & \max(Y_0, Y_1^+) \leq y \leq Y_0 + Y_1; \\ 0 & y > Y_0 + Y_1^+. \end{cases}$$

So, for $\max(Y_0, Y_1^+) \leq y < Y_0 + Y_1^+$, this leads to a probability distribution function of

$$\begin{aligned} \Pr(Y^+ = y | Y_0, Y_1^+) &= \frac{Y_0! Y_1^+!}{(Y_0 + Y_1^+ - y)!(y)!} - \frac{Y_0! Y_1^+!}{(Y_0 + Y_1^+ - (y+1))!(y+1)!} \\ &= \frac{Y_0! Y_1^+! (2y + 1 - Y_0 - Y_1^+)}{(Y_0 + Y_1^+ - y)!(y+1)!}. \end{aligned} \tag{6.37}$$

For the end point, $y = Y_0 + Y_1^+$, the probability is found from

$$\begin{aligned} \Pr(Y^+ = Y_0 + Y_1^+ | Y_0, Y_1^+) &= \frac{\binom{Y_0+Y_1^+}{Y_0+Y_1^+}}{\binom{Y_0+Y_1^+}{Y_0}} \\ &= \frac{Y_0! Y_1^+!}{(Y_0 + Y_1^+)!} \\ &= \frac{Y_0! Y_1^+! (2y + 1 - Y_0 - Y_1^+)}{(Y_0 + Y_1^+ - y)!(y+1)!}. \end{aligned} \tag{6.38}$$

so the formula of Eqn. 6.37 is valid for all y such that $\Pr(Y^+ = y) > 0$, namely $\max(Y_0, Y_1^+) \leq y \leq Y_0 + Y_1^+$. Similarly, the lower bound is found from

$$\begin{aligned} \Pr(Y^- = y | Y_0, Y_1^-) &= \frac{Y_0! Y_1^-!}{(Y_0 + Y_1^- - y)!(y)!} - \frac{Y_0! Y_1^-!}{(Y_0 + Y_1^- - (y-1))!(y-1)!} \\ &= \frac{Y_0! Y_1^-! (Y_0 + Y_1^- + 2 - 2y)}{(Y_0 + Y_1^- - y + 1)!(y)!}. \end{aligned} \tag{6.39}$$

for $0 \leq y \leq \min(Y_0, Y_1^-)$.

6.3.3 Determination of Reception Event Probabilities

Using the state variables X and R, and the auxiliary variables, the probabilities of the events described in §6.3.1 can be determined. However, the events actually depend upon the evolution of these auxiliary variables over time. Therefore, the bounds and approximations of the auxiliary variables during the interval $(0, T_p)$, described in the previous section, are used to determine bounds and approximations to the desired probabilities.

6.3.3.1 Idle Destination, P_{DI}

The probability that the destination is idle is found by counting those radios, excluding the source of the tagged packet, which are not idle at $t = 0$, which is $X + R$, plus the number that begin transmission in $(0, T_p)$, which is Y_1. Since the destination is chosen uniformly over $M - 1$ radios, P_{DI} is equal to

$$P_{DI} = 1 - \frac{X + R + Y_1}{M - 1}.$$

The bounds on Y_1 are

$$P_{DI}^- \leq P_{DI} \leq P_{DI}^+, \tag{6.40}$$

where

$$P_{DI}^- = 1 - \frac{X + R + Y_1^+}{M - 1}, \tag{6.41}$$

and

$$P_{DI}^+ = 1 - \frac{X + R + Y_1^-}{M - 1}. \tag{6.42}$$

For the approximate analysis, Y_1 is not calculated, so the unconditional probability of the destination not being idle is simply the probability that it is idle at $t = 0$ times the probability that it does not begin transmission before $t = T_p$. This is

$$P_{DI} = \left(1 - \frac{X+R}{M-1}\right) e^{-\lambda T_p}. \tag{6.43}$$

6.3.3.2 Detection and Verification, P_{d_1} and P_{d_2}

The function $P_{d_1}(I)$ is monotonically decreasing with increasing I as long as $P_{d_1}(0)$ is greater than 0.5, but is monotonically increasing with increasing I for $P_{d_1}(0) < 0.5$, and similarly for P_{d_2}. A simple analogy is to consider the c.d.f. of the tail of a family of Gaussian distributions, with the same mean but differing variances. If the mean is above the threshold, then increasing the variance will decrease the probability of being above the threshold. However, if the mean is below the threshold, the behavior is reversed. Nevertheless, from the numerical results, it is found that these probabilities are often greater than 0.5 in the range of interest, in which case upper and lower bounds on P_{d_1} and P_{d_2} can be determined from the lower and upper bounds, respectively, of the interference I during the preamble as

$$P_{d_1}(I^+) \le P_{d_1} \le P_{d_1}(I^-) \qquad \text{and} \qquad P_{d_2}(I^+) \le P_{d_2} \le P_{d_2}(I^-). \tag{6.44}$$

Furthermore, the approximations to P_{d_1} and P_{d_2} found from an approximation to I are valid even when these probabilities are less than one half, so

$$P_{d_1} \approx P_{d_1}(\hat{I}) \qquad \text{and} \qquad P_{d_2} \approx P_{d_2}(\hat{I}). \tag{6.45}$$

6.3.3.3 Interference, I

The maximum and minimum levels of interference occurring throughout $(0, T_p)$ follow directly from the auxiliary variables. Furthermore, this evaluation distinguishes between those packets which are in the data portion and those which are still in the preamble portion, allowing a more general model of the synchronization performance than that of §6.2 to be used. The upper bound on I is

$$I^+ = X - Y_0 + Y^+. \tag{6.46}$$

The term $X - Y_0$ counts all of the ongoing packets which are in the data portion at time $t = 0$. Some of these may finish before T_p, and will be counted again in Y^+, if they begin another transmission. Nevertheless, this is a small probability event, and only loosens the bound. The lower bound is

$$I^- = X - \tilde{X} - Y_0 + Y^-. \tag{6.47}$$

For random coding, there is no difference between the effect of overlapping preambles and overlapping data portions. If the preamble is small, the probability of a network transition during $(0, T_p)$ is also small, so I can be approximated as being constant and equal to X throughout the preamble.

$$\hat{I} = X. \tag{6.48}$$

X is used rather than $X - 1$ because the tagged packet is not counted in the initial total of X.

6.3.3.4 False Detection Process, λ_{fa_1}

False detections are modeled as being generated from a Poisson process. This is an approximate model, but can be justified as follows. The signal $\mathcal{Z} = \sqrt{\mathcal{Z}_I^2 + \mathcal{Z}_Q^2}$ is

compared to a threshold, $1/\xi$ times per chip. Thus, in the absence of a signal with the correct time shift, i.e., hypothesis H_0, successive correlations are over intervals which overlap to a great extent. For example, the in-phase signal $\mathcal{Z}_I(t)$ at times t_1 and $t_1 + \xi T_c$ is

$$\mathcal{Z}_I(t_1) = \int_{t_1 - T_{p_1}}^{t_1} r(s) a_0(s) \sqrt{2} \cos(\omega_c s) ds;$$

$$\mathcal{Z}_I(t_1 + \xi T_c) = \int_{t_1 + \xi T_c - T_{p_1}}^{t_1 + \xi T_c} r(s) a_0(s) \sqrt{2} \cos(\omega_c s) ds. \tag{6.49}$$

so if $\xi < 1$ the two samples of $\mathcal{Z}_I$ could be highly correlated. However, it will be shown that if ξ is an integer the two samples are nearly independent. When the sampling rate is greater than once per chip, a large number of false alarms might occur during a single chip. For a single active correlator circuit, the false alarms after the first one will have no effect, as the only active correlator will be busy. However, with multiple correlators, many could be tied up by a single noise spike. To avoid this, detections are blocked for an interval of one chip following any detection in the multiple correlator system, and the detection time is chosen as the time of the largest signal that is sampled in this interval of one chip. With this modification, there will never be more than one detection per chip, so false detections will be nearly independent. Because the chip time is small, the false detection process can be closely approximated as a Poisson process, with a rate $\lambda_{fa_1} = P_{fa_1}/T_c$.

In order to show that the two samples are independent, consider the case of constant interference, for which $\mathcal{Z}_I(t)$ can be written as

$$\mathcal{Z}_I(t) = T_{p_1} \sqrt{S} \sum_{k=0}^{I} A_k(\tau_k) \cos(\phi_k) + \mathcal{N}_I(t), \tag{6.50}$$

where the term $A_k(\tau_k)$ now explicitly indicates the dependence upon the offset τ_k, which is itself equal to some offset minus the time t. For random codes, $A_k(\tau_k)$ and

$A_k(\tau_k + \xi T_c)$ are independent when ξ is an integer. Also, the term $\mathcal{N}_I(t)$ is

$$\begin{aligned}\mathcal{N}_I(t) &= \sqrt{2}\int_{t-T_{p_1}}^{t} a_0(s)\cos(\omega_c s)n(s)ds \\ &= \sum_{j=1}^{k_1} n_j, \end{aligned} \tag{6.51}$$

where the n_j are zero-mean Gaussian r.v.'s with variance $T_c N_0/2$. At two times separated by an integer number of chips, t_1 and $t_2 = t_1 + \xi T_c$, the noise terms are

$$\begin{aligned}\mathcal{N}_I(t_1) &= \sum_{j=1}^{k_1} a_{(0,j)} n_j \\ &= \sum_{j=1}^{\xi} a_{(0,j)} n_j + \sum_{j=\xi+1}^{k_1} a_{(0,j)} n_j; \\ \mathcal{N}_I(t_2) &= \sum_{j=\xi+1}^{\xi+k_1} a_{(0,j-\xi)} n_j \\ &= \sum_{j=\xi+1}^{k_1} a_{(0,j-\xi)} n_j + \sum_{j=k_1+1}^{\xi+k_1} a_{(0,j-\xi)} n_j. \end{aligned} \tag{6.52}$$

Because the values of n_j are independent for different j, the non-overlapping portion of the sums in Eqn. 6.52 are independent. If ξ is less than k_1, the overlap is non-zero, and is given by the terms

$$\sum_{j=\xi+1}^{k_1} a_{(0,j)} n_j \qquad \text{and} \qquad \sum_{j=\xi+1}^{k_1} a_{(0,j-\xi)} n_j. \tag{6.53}$$

Let Υ^+ to be those positions $j \in [\xi+1, k_1]$ for which $a_{(0,j)} = a_{(0,j-\xi)}$, and let Υ^- be those positions for which $a_{(0,j)} = -a_{(0,j-\xi)}$. For random codes, unless $k_1 - \xi$ is very small, in which case the overlap will be practically negligible, the sizes of the sets Υ^+ and Υ^- will be very nearly equal to the same value of $(k_1 - \xi)/2$. Defining U_+ and U_- as

$$U_+ = \sum_{j\in\Upsilon^+} a_{(0,j)} n_j \qquad \text{and} \qquad U_- = \sum_{j\in\Upsilon^-} a_{(0,j)} n_j. \tag{6.54}$$

the overlapping sums of $\mathcal{N}_I(t_1)$ and $\mathcal{N}_I(t_2)$ are

$$\sum_{j=\xi+1}^{k_1} a_{(0,j)} n_j = U_+ + U_- \qquad \text{and} \qquad \sum_{j=\xi+1}^{k_1} a_{(0,j-\xi)} n_j = U_+ - U_-. \tag{6.55}$$

But U_+ and U_- are zero mean Gaussian random variables, so the sum and difference $U_+ + U_-$ and $U_+ - U_-$ are also zero mean Gaussian r.v.'s. Furthermore, the covariance between the sum and difference is

$$E((U_+ + U_-)(U_+ - U_-)) = E(U_+^2) - E(U_-^2). \tag{6.56}$$

However, if $\|\Upsilon^+\| \approx \|\Upsilon^-\|$ then $E(U_+^2) \approx E(U_-^2)$, so the covariance is nearly zero. Gaussian random variables that are uncorrelated are also independent, so to a good approximation, the two sums $\mathcal{N}_I(t_1)$ and $\mathcal{N}_I(t_2)$ are independent. Thus, because both the interference terms and the noise terms of $\mathcal{Z}_I(t_1)$ and $\mathcal{Z}_I(t_1 + \xi T_c)$ are approximately independent, these two terms are also approximately independent. A similar argument holds for the quadrature terms $\mathcal{Z}_Q(t_1)$ and $\mathcal{Z}_Q(t_1+\xi T_c)$, so the false alarms at times separated by an integral number of chips are nearly independent. Therefore, the false alarms can be accurately modeled as generated from a Poisson process.

6.3.3.5 Not Busy Verifying, P_{NV}

The probability that the active correlator is busy at time T_{p_1}, given that the destination is idle at $t = 0$, given that the preamble was detected, and given the various state and auxiliary variables, has been defined as P_{NV}. For several reasons, an exact determination of P_{NV} from the network state and the auxiliary variables is found to be intractable. The event of a busy active correlator depends upon both false detections and true detections during the interval $(T_{p_1} - T_{p_2}, T_{p_1})$. These detections

are a function of the interference occurring throughout the interval $(-T_{p_2}, T_{p_1})$. The first difficulty is that the evolution of the network over this interval is not precisely known. Second, for both true and false detections, it is not enough to know if a detection has occurred, because the active correlator may have been busy at the time of this earlier detection. Thus, the state of the active correlator at time T_{p_1} is a function of both the detections occurring in $(T_{p_1} - T_{p_2}, T_{p_1})$ and the state of the active correlator at the times of those detections. The third difficulty arises because P_{NV} is conditioned on the destination being idle at time $t = 0$. By definition, this means that it does not acquire any of the preambles in V. This condition has a large impact on the probability that the destination was verifying an overlapping preamble, as the two events are highly correlated. For example, in the extreme case of $P_{d_2} = 1.0$, if the destination is idle then it cannot be verifying an overlapping preamble, since any preamble which is being verified will be acquired, in which case the destination would have been counted as receiving rather than idle at time $t = 0$. Because an exact determination is not possible, bounds on and approximations to P_{NV} are determined.

The effect of the interference during $(-T_{p_2}, T_{p_1})$ on the overall probability of reception is a second order effect. Therefore, the analysis is based on an approximation to this interference. The small inaccuracies due to this approximation should have little effect on the final result. The interference is assumed to be constant, and equal to $\hat{I}_0$, where

$$\hat{I}_0 = X. \tag{6.57}$$

X is used rather than $X - 1$ because the source is not included in X.

For the lower bound on P_{NV}, it is assumed that any detection, whether false or not, finds an idle correlator. Thus, the only way that the correlator can be free at time T_{p_1} is if there are no detections in $(T_{p_1} - T_{p_2}, T_{p_1})$. With this bound, the effects

of false detections and of true detections become uncoupled, since the probabilities of detection are themselves independent when conditioned on the variables and the parameter settings. Therefore, the bound P_{NV}^{-} can be written as

$$P_{NV}^{-} = P_{NVF}^{-} P_{NVT}^{-}. \tag{6.58}$$

A tight upper bound has not been determined. However, the simple bound $P_{NV}^{+} = 1$ can be used. Of course, this bound will become loose when the probability of false detection becomes large, but at the optimum operating point, this probability is small, so the bound will be useful.

For the approximation, it is assumed that the probability of an overlapping preamble is negligible, so that $\hat{P}_{NVT} = 1$, regardless of the number of overlapping preambles on the same code. There are two justifications for choosing this approximation. First, if the preamble is small, which is often the case, the probability that V is not zero will be small, so the values of P_{NVT} conditioned on V non-zero will have very little effect on the overall result. Second, even when V is not zero, it is believed that the condition that the destination is idle greatly reduces the probability of the event that one of the overlapping preambles is verified. With the approximation $\hat{P}_{NVT} = 1$, the probability that the active correlator is free is equal to the approximate probability that no false detections are being verified.

$$\hat{P}_{NV} = \hat{P}_{NVF}. \tag{6.59}$$

The quantity $\hat{P}_{NVF}$ is derived in §6.3.3.6.

6.3.3.6 Not Verifying False Detections, P_{NVF}

The lower bound on P_{NVF} is found by assuming the worst case, which is that any false detection arriving in the interval $(T_{p_1} - T_{p_2}, T_{p_1})$ finds the active correlator

idle. With this assumption, P_{NVF} is simply the probability of no arrivals from a Poisson process, rate λ_{fa_1}, in time T_{p_2}, so

$$P_{NVF}^{-} = e^{-\lambda_{fa_1} T_{p_2}} = e^{-P_{fa_1}(\hat{I}_0) k_2}. \tag{6.60}$$

An approximation to P_{NVF}, $\hat{P}_{NVF}$, is found from considering a steady state system, in which λ_{fa_1} is constant over all time, and the effect of verifying true preambles is ignored. The resulting system is simply a single server queue, with a fixed service time of T_{p_2} and Poisson arrivals, rate λ_{fa_1}. The probability that a correlator is not verifying a false detection is thus

$$\hat{P}_{NVF} = 1 - \frac{T_{p_2}}{T_{p_2} + 1/\lambda_{fa_1}} = \frac{1}{P_{fa_1}(\hat{I}_0) k_2 + 1}. \tag{6.61}$$

6.3.3.7 Not Verifying Overlapping Preambles, P_{NVT}

As in the analysis of P_{NVF}, in the worst case it is assumed that any of the V overlapping preambles which is detected finds the active correlator idle. Furthermore, the conditioning on P_{DI} increases P_{NVT}, so for the worst case, P_{NVT} is considered to be independent of P_{DI}. For this case, P_{NVT} is simply the probability that none of the V overlapping preambles is detected,

$$P_{NVT}^{-} = \left(1 - P_{d_1}(\hat{I}_0)\right)^V. \tag{6.62}$$

For both the approximation and the upper bound, it is assumed that no preambles are verified, so $\hat{P}_{NVT} = P_{NVT}^{+} = 1$.

6.3.3.8 False Lock, P_{FL}

The final event which must be considered is the possibility of the destination being in false lock at time T_p, conditioned on all the previously discussed events being successful. Before the probability of false lock can be analyzed, the effect of a false lock must first be specified. For simplicity, in this analysis it is assumed that each false lock occupies the destination for a fixed length of time, T_{fl}, which is equal to k_{fl} chips. Numerical results are found for T_{fl} corresponding to 64 bits.

In order to determine whether a radio is in false lock, it is necessary to know the network states over the last 64 bits, as P_{fa_1} and P_{fa_2} must be calculated for the network state in the interval of length T_p preceding the time of the start of false lock. Typically, the preamble will only be several bits long, so the time scale needed to determine false lock is different from that used in the earlier analysis. In previous chapters, it has been shown that the network state changes slowly over tens of bits. The probability of false lock is calculated by making the loose approximation that the state was constant over the entire 64 bits. Although this is only an approximation, the probability of false lock is small, so the error due to this approximation should have little impact on the overall throughput. The interference level $I_{fl} = X - 1$ is used, since the tagged preamble is only a small fraction of the interval of 64 bits. In the worst case, every false detection finds the active correlator idle, so the probability of no false lock, $1 - P_{FL}$, is approximately equal to the probability of no arrivals from a Poisson process, rate $P_{fa_2}\lambda_{fa_1}$ in a time T_{fl}. Therefore,

$$1 - P_{FL} \approx \exp\{-P_{fa_1}(I_{fl})P_{fa_2}(I_{fl})k_{fl}\}. \tag{6.63}$$

6.3.4 Bounds on the Probability of Reception

Given the various constituent probabilities of §6.3.3, overall bounds on the probability of reception can now be found. As discussed in §6.3.3.2, these overall bounds are only valid when the probabilities of detection P_{d_1} and P_{d_2} are greater than 1/2. The lower bound is determined using the worst case results, giving

$$P_{Rx}(X,R) \geq \sum_{y_0=0}^{X} \Pr(Y_0 = y_0) \sum_{v=0}^{Y_0} \Pr(V = v) \sum_{x=0}^{X} \Pr(\tilde{X} = x) \times$$
$$\sum_{r=0}^{\min(R,\tilde{X})} \Pr(\tilde{R} = r) \sum_{y_1=0}^{J} \Pr(Y_1^+ = y_1) \sum_{y=\max(Y_0,Y_1^+)}^{Y_0+Y_1^-} \Pr(Y^+ = y) \times$$
$$P_{DI}^- P_{d_1}(I^+) P_{NV}^- P_{d_2}(I^+)(1 - P_{FL}). \quad (6.64)$$

The upper bound is

$$P_{Rx}(X,R) \leq \sum_{y_0=0}^{X} \Pr(Y_0 = y_0) \sum_{y_1=0}^{J^-} \Pr(Y_1^- = y_1) \sum_{y=0}^{\min(Y_0,Y_1^-)} \Pr(Y^- = y) \times$$
$$P_{DI}^+ P_{d_1}(I^-) P_{d_2}(I^-)(1 - P_{FL}). \quad (6.65)$$

As will be seen in §6.4, the maximum throughput is achieved with a preamble length of only several bits. For such a small preamble, the probability of network transitions is practically negligible. Therefore these bounds can be closely approximated by simpler expressions which do not account for all of the possible network transitions during the preamble.

The approximate lower bound is

$$P_{Rx}(X,R) \gtrsim \sum_{v=0}^{X} \Pr(V = v) P_{DI} P_{d_1}(\hat{I}) P_{NV}^- P_{d_2}(\hat{I})(1 - P_{FL}). \quad (6.66)$$

The approximate upper bound is

$$P_{Rx}(X,R) \lesssim P_{DI} P_{d_1}(\hat{I}) P_{d_2}(\hat{I})(1 - P_{FL}). \quad (6.67)$$

6.3.5 Approximation to the Probability of Reception

For small preamble times, a closer approximation to the true performance is found by making use of the approximations of §6.3.3, rather than the bounds. In addition to giving closer agreement to the results from simulation, the approximation is also a simpler expression,

$$P_{Rx}(X,R) \approx P_{DI}P_{d_1}(\hat{I})\hat{P}_{NV}P_{d_2}(\hat{I})(1-P_{FL}). \tag{6.68}$$

6.3.6 Multiple Active Correlators

The analysis of a system with c active correlators is similar to the $c = 1$ case except for the probability of not verifying, P_{NV}, which will now denote the probability that at least one active correlator is available.

For the lower bound, it is again assumed that any detection will tie up a correlator. Thus, P_{NV}^{-} is found as the probability that the number of active correlators verifying true signals (which is binomially distributed) plus the number verifying false signals (which is Poisson distributed) is less than c,

$$P_{NV}^{-} = \sum_{i=0}^{V} \binom{V}{i} (P_{d_1}(\hat{I}_0))^i (1-P_{d_1}(\hat{I}_0))^{V-i} \sum_{j=0}^{c-i-1} \frac{(P_{fa_1}(\hat{I}_0)k_2)^j e^{-P_{fa_1}(\hat{I}_0)k_2}}{j!}. \tag{6.69}$$

For the upper bound and approximate upper bound, there is no change, as the bound $P_{NV}^{+} = 1$ is used. For the approximation, it is again assumed that $P_{NV|T} = 1$, and so $\hat{P}_{NV}$ can be found from the Erlang B formula, $\hat{P}_{NV} = B(c,a)$, where the normalized arrival rate $a = P_{fa_1}(\hat{I}_0)k_2$. Numerically, this is computed from the recursion $B(c,a) = [aB(c-1,a)]/[c + aB(c-1,a)]$, as in [44]. Therefore, the approximations and bounds found earlier are easily extended to the case of a system with multiple active correlators.

6.3.7 Multiple Cycles

The case of a system with multiple cycles is analyzed by making the approximation that product $P_{d_1}P_{d_2}P_{NV}$ is independent from cycle to cycle. The detection probabilities will be very nearly independent. However, the probabilities P_{NV} are not independent. Nevertheless, as seen from numerical results, P_{NV} will be close to one for most regions of interest, so the dependence from cycle to cycle will not have a great effect on performance. This independence assumption has been found to give good agreement with results from simulation which take into account the dependence. A further change is in the determination of P_{DI}. Because a packet can be acquired before the end of the preamble, the probability that the destination begins transmitting during a given cycle is conditioned on the packet not yet being acquired. The probability that the preamble is acquired on the mth cycle is

$$P_{Rx}^{(m)} = \left(1 - \frac{X+R}{M-1}\right) e^{-(m\lambda T_p/n)}(1 - P_{d_1}P_{NV}P_{d_2})^{m-1}P_{d_1}P_{NV}P_{d_2}. \qquad (6.70)$$

so the approximations to the bounds and the approximation for a system with n cycles per preamble are found from the Equations 6.66, 6.67, and 6.68 by replacing the quantity $P_{DI}P_{d_1}P_{NV}P_{d_2}$ with

$$\sum_{m=1}^{n} P_{Rx}^{(m)}. \qquad (6.71)$$

6.4 Synchronization Results

In order to find the overall best performance, it is necessary to maximize over a large number of variables. For most of the results, the length of the matched

filter, k_1 is constrained, and the throughput is maximized over l_1, k_2, l_2, and g. Brent's method for global minimization [8] is used to numerically find the maximum throughput, by minimizing $-S$, the negative of the the throughput. All numerical results given in this section are for the coded channel, with an offset $|\tau_0| = 0.25$, and $\eta = 1.0$.

The first plot is a comparison of the results for ideal synchronization to the realistic capacity for a system using the two stage synchronization circuit. Also shown is the capacity resulting from a simpler synchronization circuit consisting of only a matched filter with no active correlator. In Fig. 6.6, the capacity C is found for these three cases for the finite population network, $M = 20$, with the coded channel, $E_b/N_0 = 8.0$ dB and 12.0 dB. The length of the matched filter is equal to 500. At low spreading factors, there is very little difference between the three cases. However, at higher spreading factors, the channel becomes better but the performance of the synchronization circuit becomes worse. This occurs because k_1, k_2, T_{bit}, and E_b/N_0 are all fixed, but T_c becomes smaller for larger N, so the energy per chip, and consequently, the energy in an interval of k_1 or k_2 chips, decreases. The degradation is more pronounced for the lower value of received power. A similar plot is given in Fig. 6.7, in which the performance of the two-stage synchronization circuit is shown for matched filter lengths of $k_1 = 500$ and 1000, again for $E_b/N_0 = 8.0$ dB and 12.0 dB.

The two operating points $N = 256$, $k_1 = 1000$, $E_b/N_0 = 8.0$ dB and $E_b/N_0 = 12.0$ dB are examined for comparing the various bounds and approximations to results from simulation. The approximate optimum values of l_1, k_2 and l_2 are found from the analytical approximation. Using these values, the throughput is found from simulation, and the bounds of §6.3.4 are also determined. The results are given in Table 6.1. For the lower value of E_b/N_0, the optimum performance is achieved with

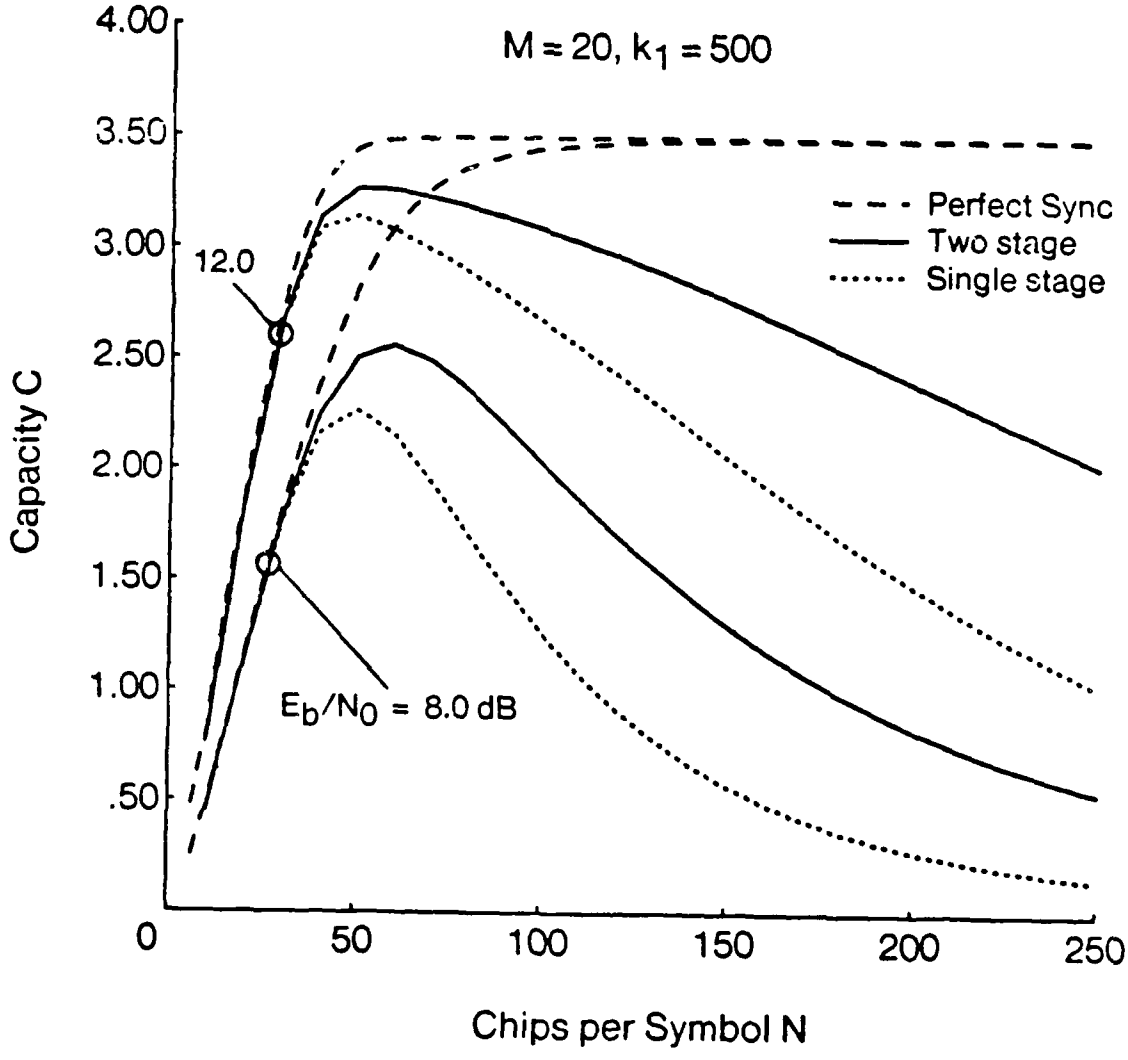


Fig. 6.6 Capacity versus spreading factor, ideal and realistic synchronization.

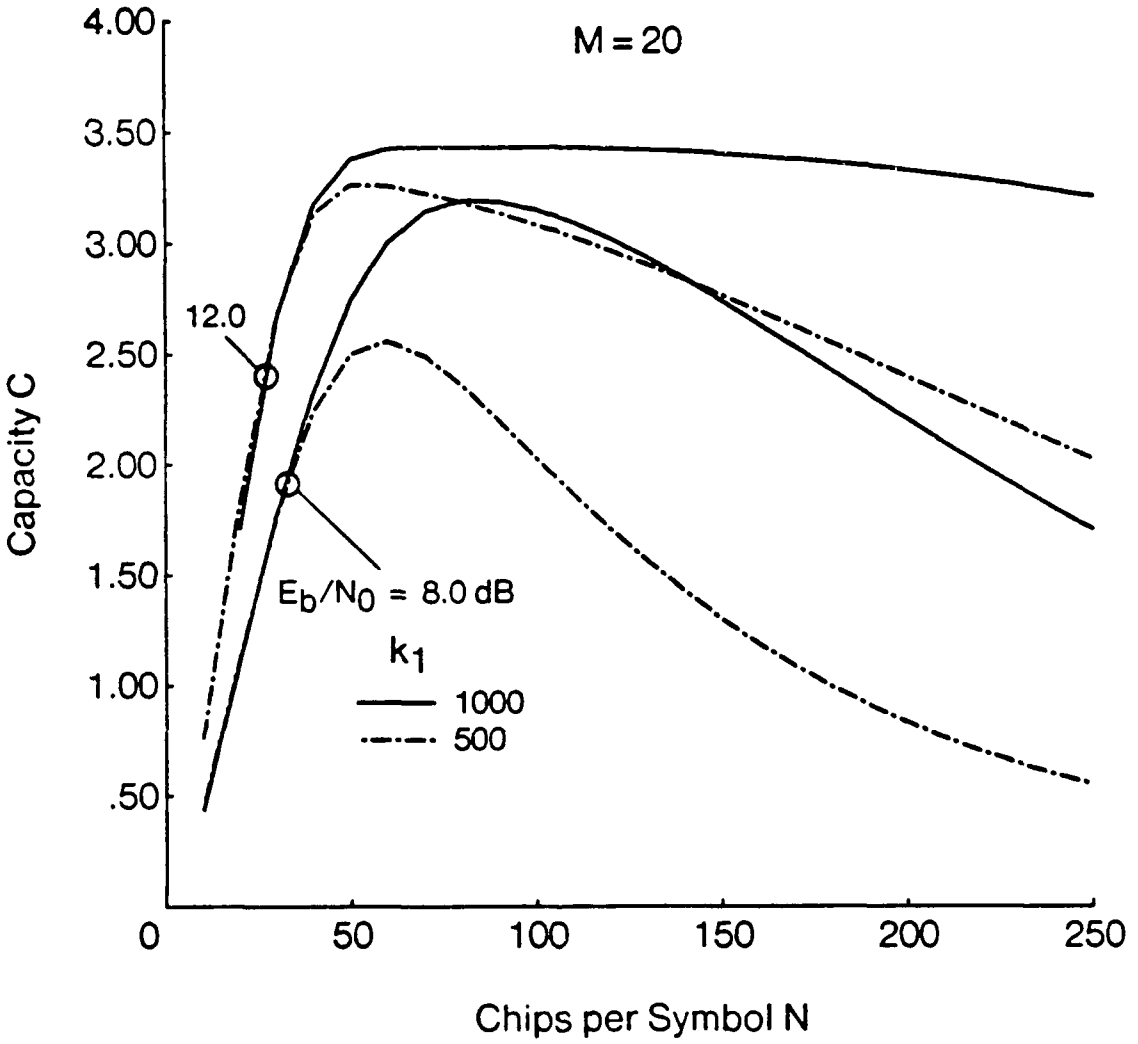


Fig. 6.7 Capacity versus spreading factor and matched filter length.

Result	8.0 dB	12.0 dB
Upper Bound	*	3.280
Upper Approximation	1.985	3.279
Lower Bound	*	3.188
Lower Approximation	1.616	3.189
Approximation	1.654	3.192
Simulation	1.67	3.23

Table 6.1. Comparison of bounds, approximations, and simulation results.

a P_{d_1} that is lower than 1/2, so the bounds are not valid. Nevertheless, for the higher E_b/N_0, it is seen that the approximations to the bounds are almost identical to the bounds themselves, and these approximations do not require on P_{d_1} to be less than 1/2. The bounds are fairly tight for the optimum parameter settings. Furthermore, the approximation is within 2% of the simulation results, for both 8.0 dB and 12.0 dB. Also, it is seen that the bounds do indeed bracket the simulation results. Therefore, at the point which gives the maximum throughput using the approximate model, the results are very close to the more exact results found from simulation.

The sensitivity to the various parameters is found for the point $N = 256$, $k_1 = 1000$, $E_b/N_0 = 8.0$ dB. The next several plots indicate the sensitivity to the parameters g, l_1, k_2, and l_2 around the optimum values. Only a single parameter is varied at a time; the parameters that are constant are set equal to the optimum values. In Fig. 6.8, the throughput S is plotted as a function of the traffic rate g. Also shown are the simulated points at the optimum, and above and below the optimum value of g. Similar plots for l_1, k_2, and l_2 are given in Figs. 6.9, 6.10 and 6.11. In general, there is good agreement between the simulated points and

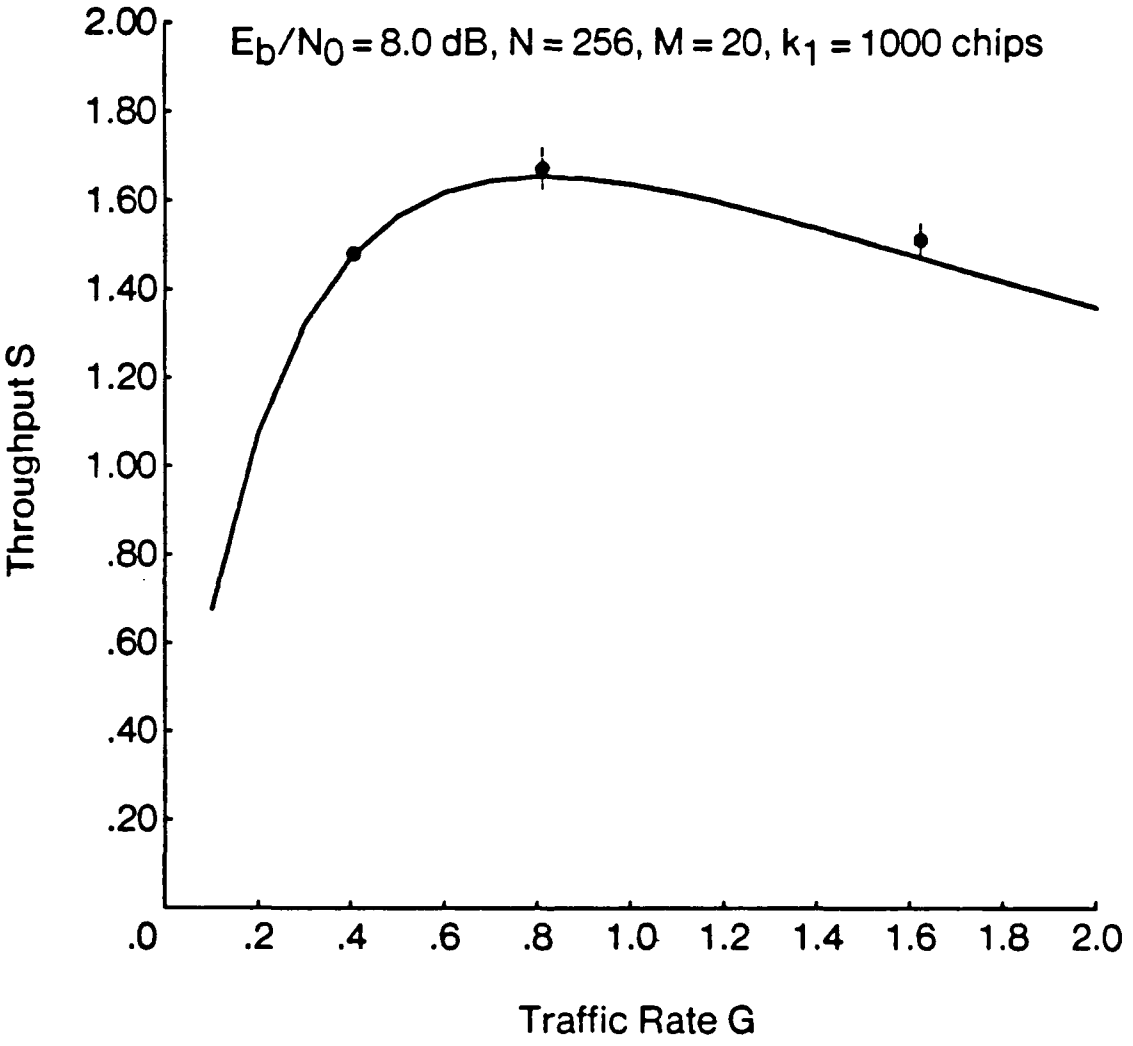


Fig. 6.8 Throughput versus traffic rate, realistic synchronization.

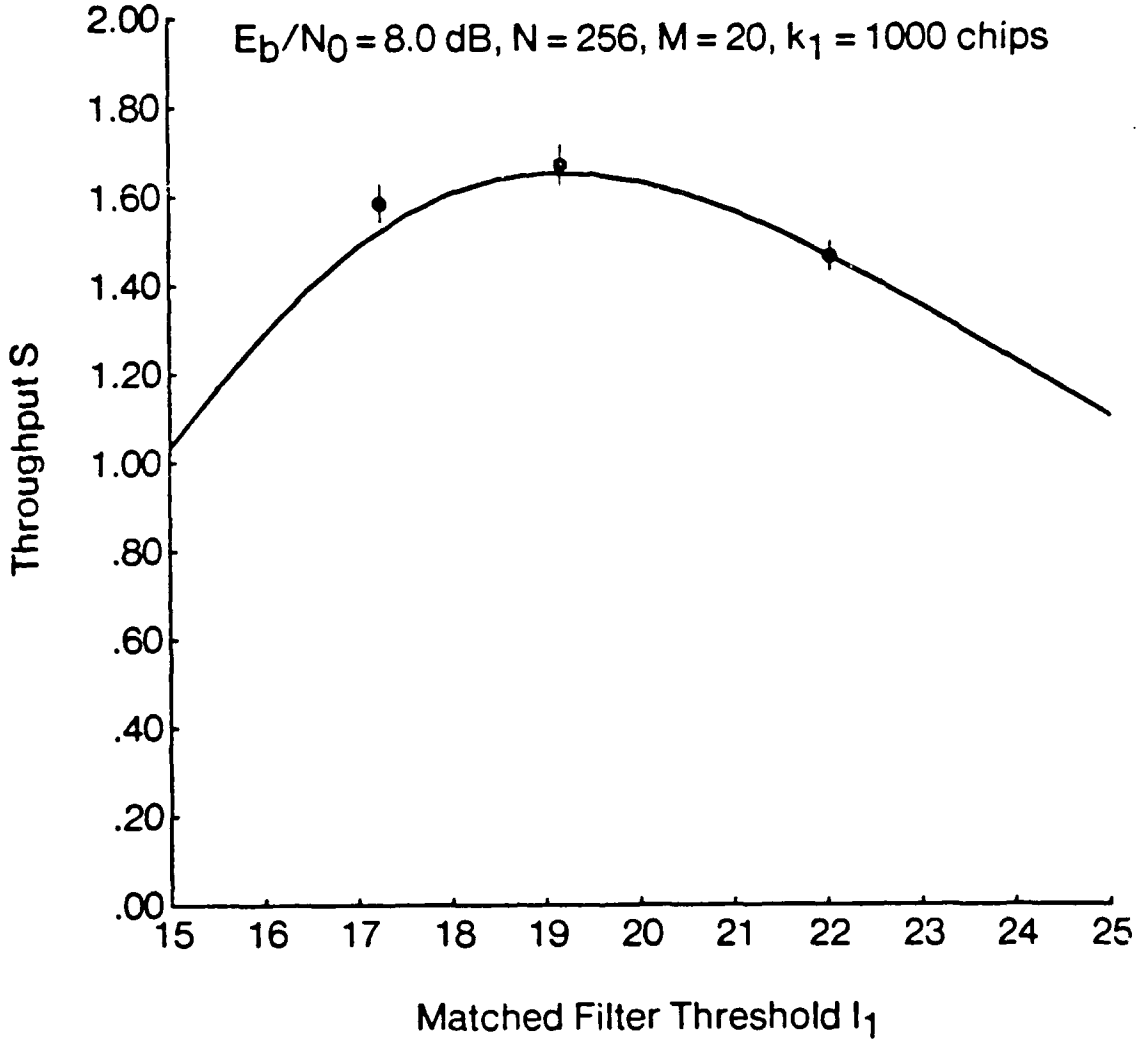


Fig. 6.9 Throughput versus matched filter threshold.

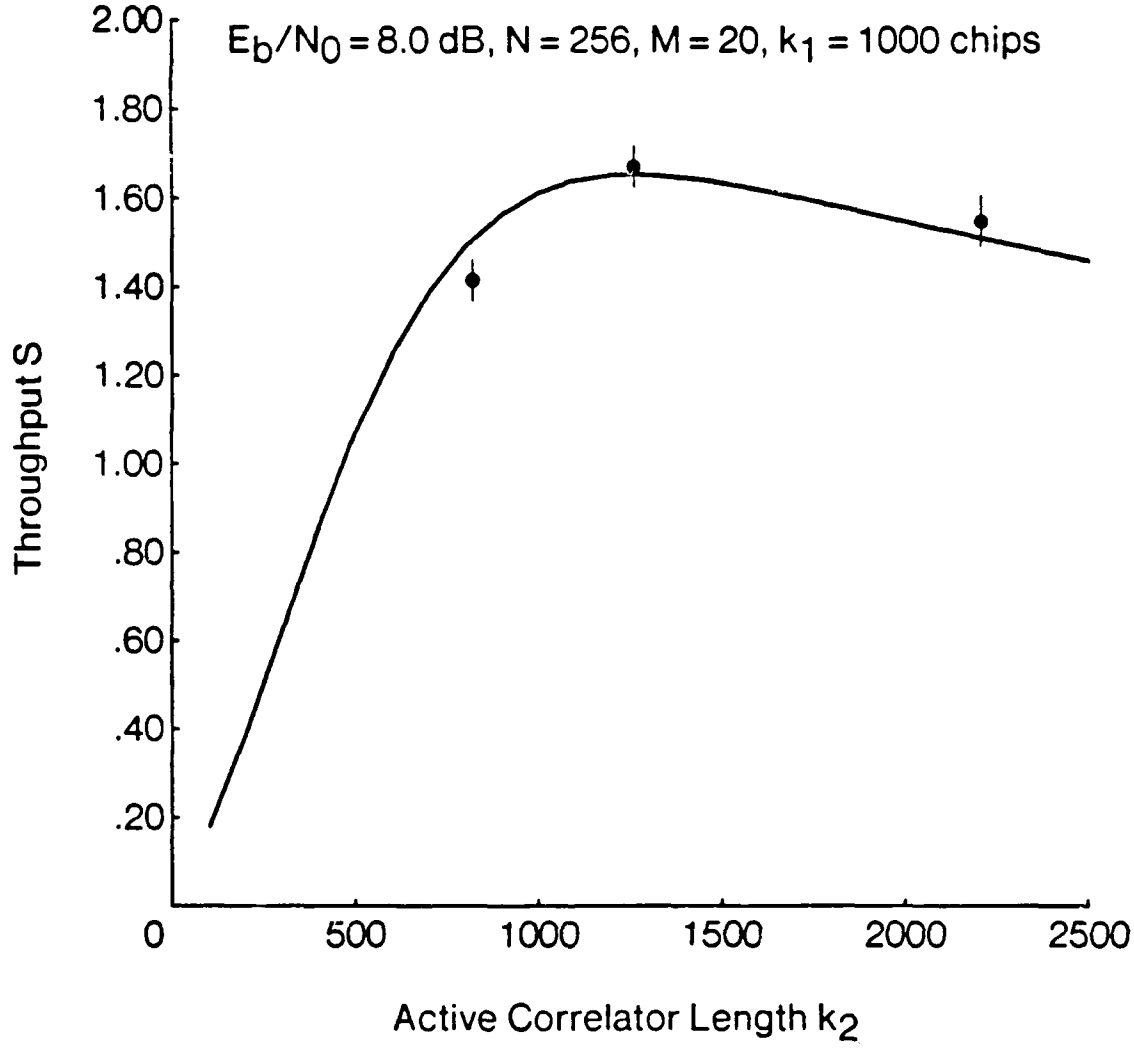


Fig. 6.10 Throughput versus active correlator length.

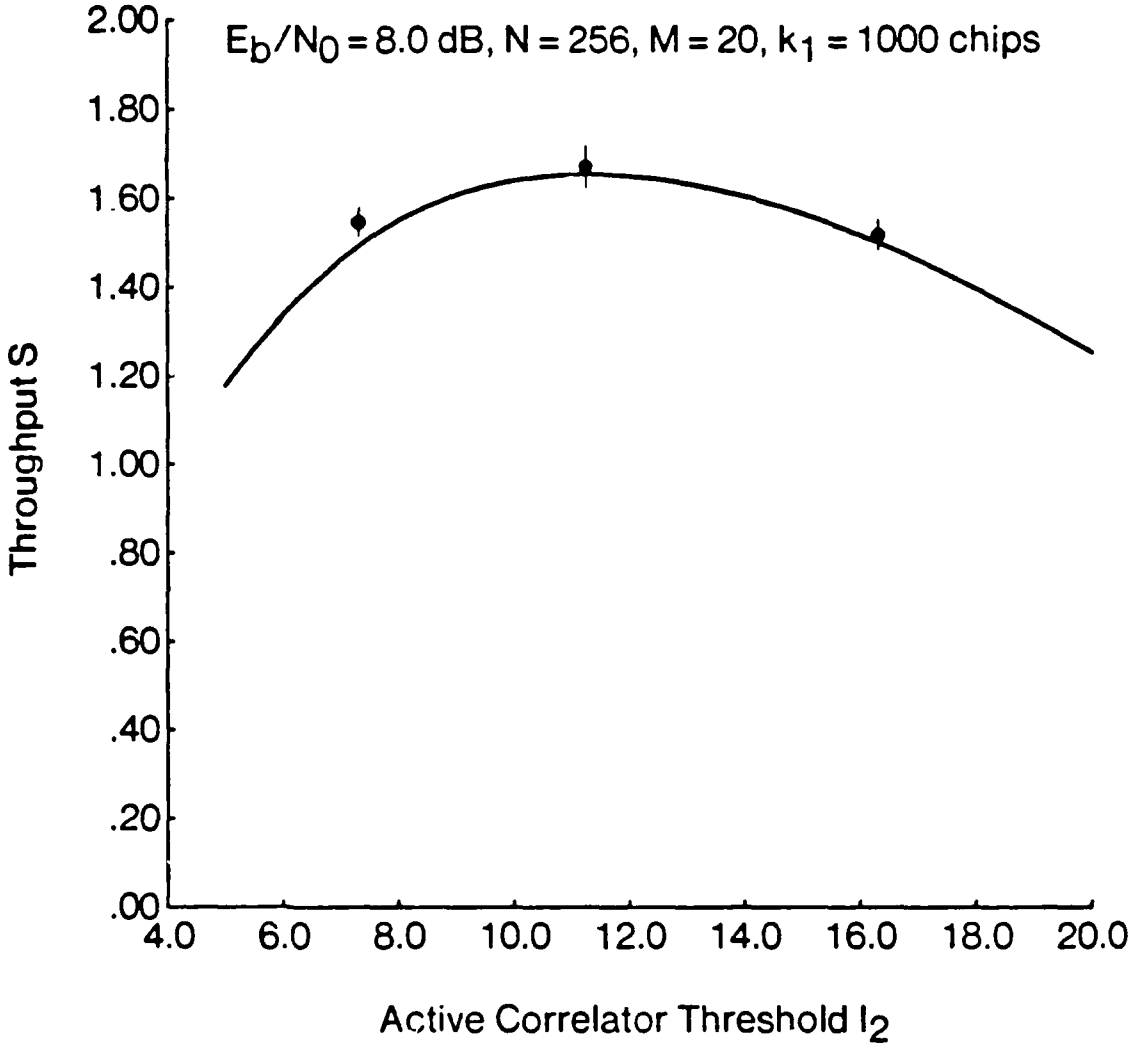


Fig. 6.11 Throughput versus active correlator threshold.

the analytical results plotted. As the parameters move away from the optimum, some of the approximations become loose. Nevertheless, the worst error between the analytical results and the simulation results is 6% for $k_2 = 820$. The true optimum values of the parameters will be in the vicinity of the values given by the approximate analysis. Furthermore, the actual capacity achieved at that optimum will be very close to the capacity indicated by the analysis. Therefore, subsequent results will be those derived from the approximation of §6.3.5.

In Fig. 6.12, the capacity is plotted as a function of the number of active correlators c, for matched filter lengths of 500 to 2000. Also shown is the simulation result for $k_1 = 500$, $c = 5$. The difference from simulation is 10%, which is not as good as in the case of a single active correlator, but is small enough that the approximate analytical results are still meaningful. The performance is greatly improved by the addition of more active correlators, and for $k_1 = 2000$, the performance comes very close to the capacity of 3.5 found with perfect synchronization. However, the addition of more active correlators leads to a more expensive synchronization circuit.

The alternative, adding more cycles per preamble, does not have the consequence of higher cost. Yet, as shown in Fig. 6.13, similar improvement can be achieved. In this figure, the capacity is plotted as a function of the matched filter length, for several values of n, the number of cycles. The simulation result for the number of cycles $n = 5$, with $k_1 = 500$ is also shown, and is found to be very close to the analytical result. For a very large n, the total length of the preamble becomes a significant fraction of the preamble, so the approximation become loose. For example, the extreme point of $n = 20$, $k_1 = 2000$ corresponds to a preamble length of about 80 bits out of an average packet length of 1000, not even including those chips that make up the second part of each cycle. Nevertheless, the results for lower values of n and k_1 are valid, and the results for longer preambles indicate

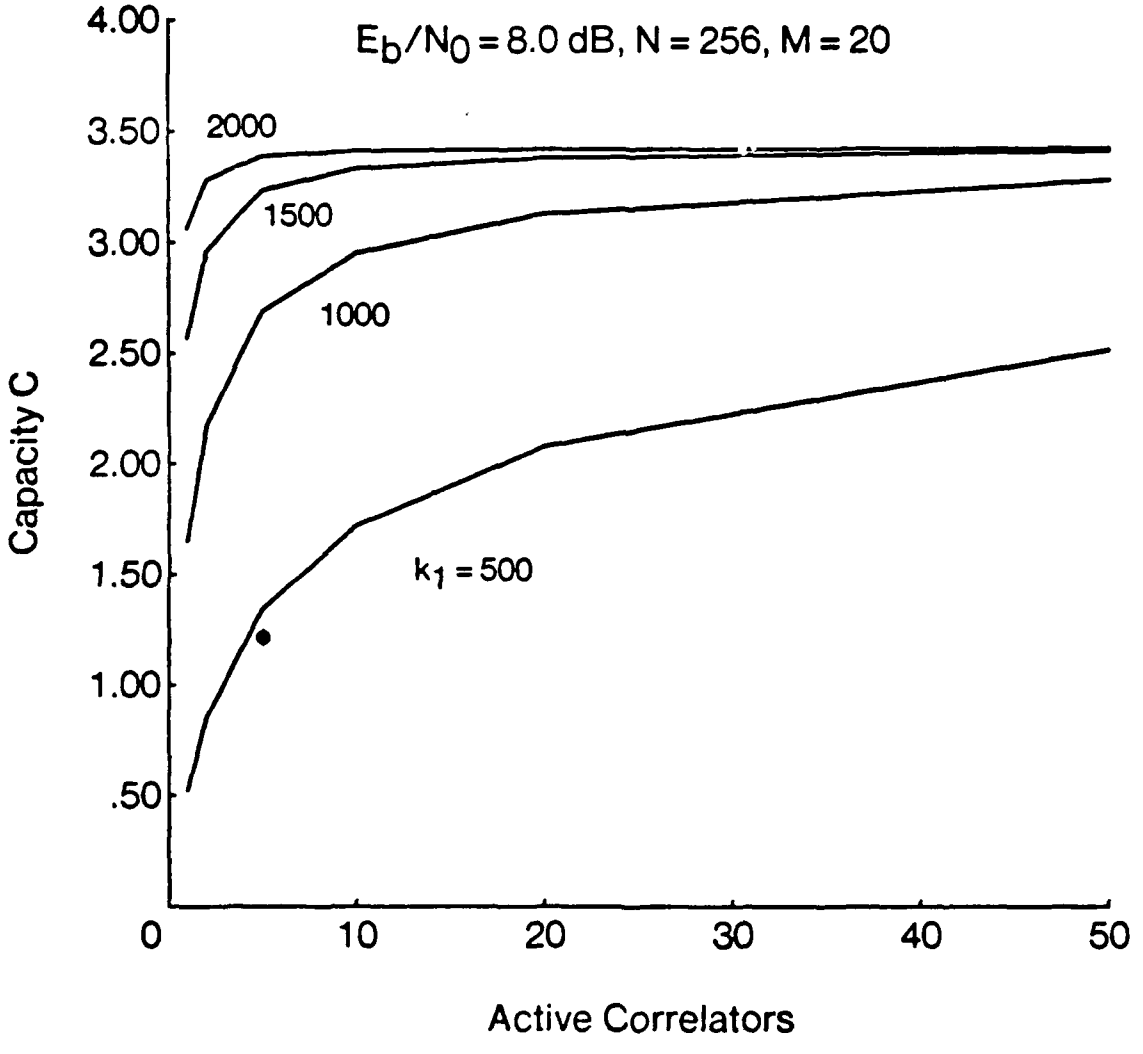


Fig. 6.12 Capacity versus number of active correlators and matched filter length.

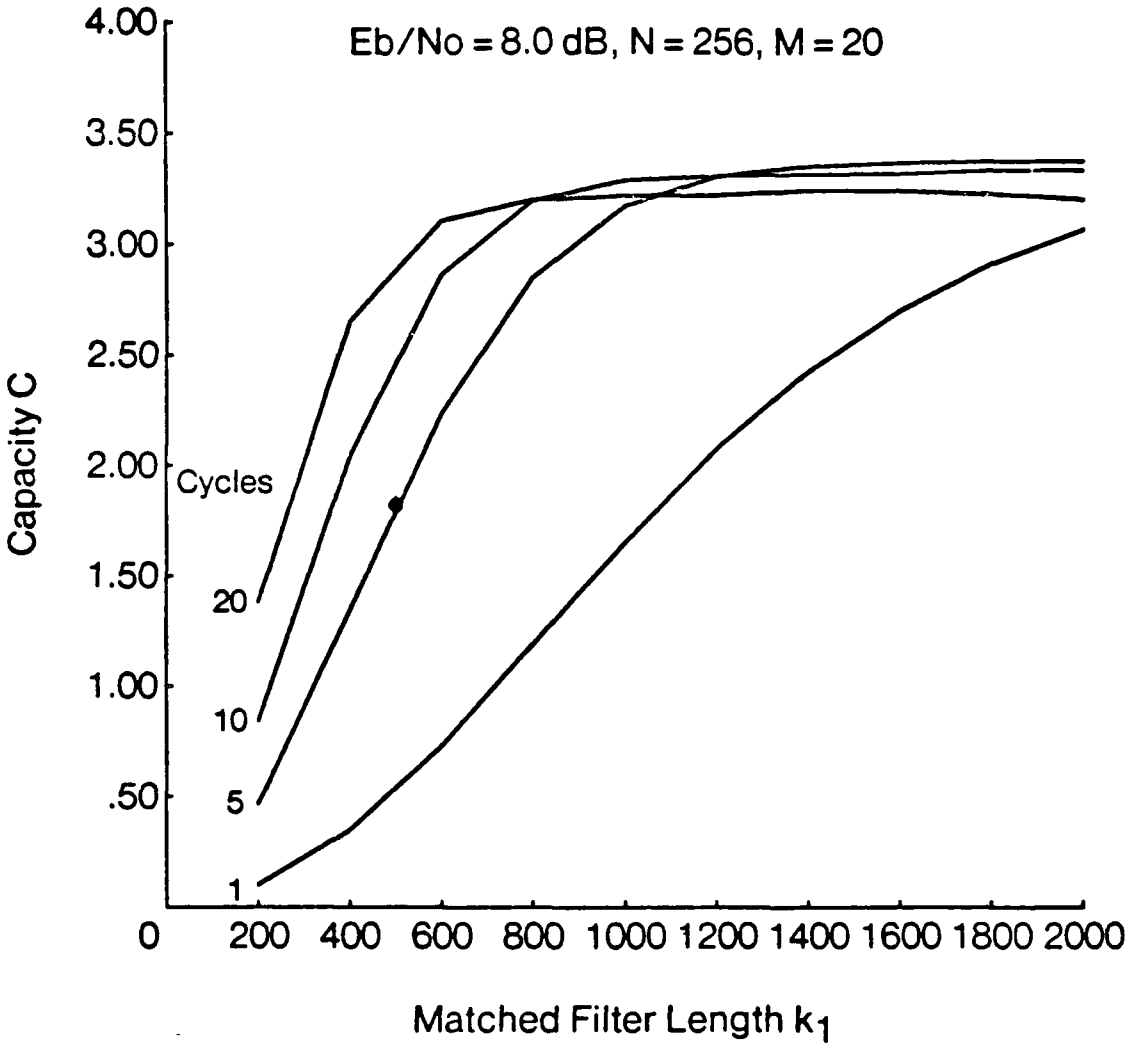


Fig. 6.13 Capacity versus matched filter length, multiple cycles.

the trends if not the actual throughputs.

In Fig. 6.14, the throughput is shown as a function of n for different matched filter lengths k_1. For a given k_1, there is an optimum value of n. For n larger than this optimum, the overhead for the preamble becomes significant. By choosing the number of cycles correctly, the synchronization performance for a matched filter of 1000 chips or more can be very close to the ideal. For the case of $k_1 = 1000$, the optimum n was found to be 9, and this resulted in a capacity of 3.29, which is 94% of the ideal capacity of 3.50.

6.5 Summary

The analysis of this chapter addressed the problem of packet acquisition, that is, detection and synchronization to the spread spectrum code. There were two tasks involved. First, the probability of detection and of false alarm were determined for the optimum I-Q detector in the environment of an SSMA PR network. By making the key approximation that the phase offsets of the received signals were all identical, the analysis led to an intermediate result from which the the work of Polydorous and Weber [36] could be followed. The second task was that of identifying and also bounding or approximating the probabilities of the various events constituting the start of a packet reception. The resulting approximation was found to agree very well with the true results from simulation, especially near the parameter settings which give the maximum throughput. The analysis was extended to a multiple correlator system and also to a system with a multiple cycle preamble.

Numerical results indicated that with the single cycle, single correlator circuit, unless k_1 is very large, the throughput becomes limited by the synchronization performance for fairly low values of chips per symbol N. At higher N, the performance

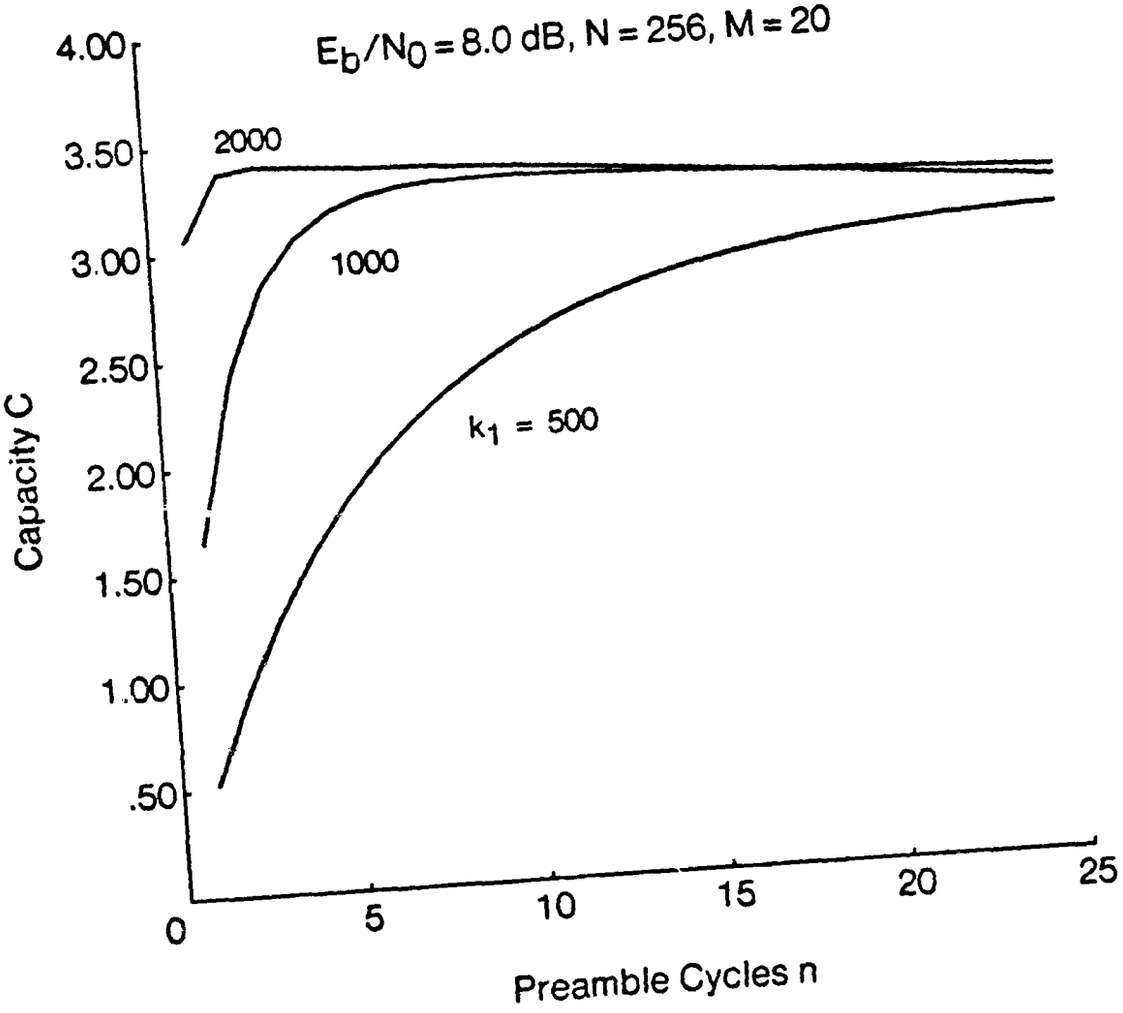


Fig. 6.14 Capacity versus number of cycles and matched filter length.

degrades significantly, especially for the moderate E_b/N_0 of 8.0 dB. However, the performance can be improved by using a multiple correlator system or a multiple cycle system. Because the former requires additional hardware, the prefered solution is to use many cycles per preamble. With a multiple cycle system, the maximum throughput achieved was found to be close to the throughput resulting from the assumption of ideal synchronization. Therefore, the results of Chapters 3 through 5 are valid approximations to the capacity of a system with a multiple cycle two stage synchronization circuit, for which the parameters n, l_1, k_2, and l_2 are optimized for each value of N, E_b/N_0, M, and g.

Chapter 7

Conclusion

The integrated model developed in this dissertation incorporates the important details of a spread spectrum multiple access packet radio network at both the channel and the network level. The final version, described in Chapter 5, is a complete and accurate model of the system. Although the derivations of the various portions that constitute the model are not all exact, the approximations are very tight in the region of maximum throughput, as verified by simulation. The results derived here are perhaps the first network level results which account for both bit errors (or decoder errors) and preamble synchronization in detail. Because the model is comprehensive, the final results are in terms of network throughput, rather than as intermediate measures such as bit error rates or detection and false alarm probabilities. This allows the impact of the channel level parameters to be judged immediately. Also, the interaction between the various aspects of the channel and the network can be determined.

The performance of the Viterbi decoder and the performance of the synchronization process were analyzed for the environment of an unslotted multiple access network. Previous results for the Viterbi decoder all rely on the assumption that the

symbol error rate is constant. Furthermore, most of the results consider the reception of a continuous stream of data, rather than a short packet. Similarly, most of the analyses of spread spectrum synchronization circuits are aimed towards finding the time to acquire a continuous stream of data. Also, there are no detailed analyses of the synchronization performance in a system with multi-user interference. Thus, before network level effects could be found, it was necessary to determine the performance of these particular areas of the system.

The aim of this research was not merely to derive an accurate theoretical model of the system, but to formulate a model which would lead to numerical results. Although there has been very little discussion of the computer programming involved, this was as large a task as the derivation of the models. In addition to the writing, this also included verifying the correctness of the programs. A modular approach was taken, and the intermediate results of sample runs were checked extensively. An outside check was provided by the results of the simulation, which agreed with the analytical results to well within the confidence intervals. Therefore, there is a very high level of confidence in the correctness of the programs and the numerical results calculated.

From the numerical results, three major effects on throughput can be identified: receiver availability, channel errors, and synchronization. The first effect, receiver availability, stems from the random starting points of the packets and the simple fact that a packet cannot be received if the destination is busy at the start of that packet. Receiver availability becomes a limitation if there is sufficient bandwidth available that the spreading W can be large; the signal power is large compared to the thermal noise density (the exact E_b/N_0 required depends upon the modulation and coding used); and the synchronization performance is close to ideal for the values of W and E_b/N_0 that are used. For this channel, the probability of error will

be small even with most of the radios transmitting, which is the perfect capture assumption, and all packets destined to idle radios will be acquired, so the only factor that can limit the throughput is the probability that the destination is busy. With a perfect channel and perfect synchronization, for $M > 10$, the network capacity is only slightly higher than the asymptotic value of $0.172M$, which occurs for $M \to \infty$.

The second effect is the presence of bit errors. For the uncoded channel, even with a very high signal power E_b/N_0 and a large number of chips per bit W, the probability of bit errors has at least some effect on throughput. For the channel with forward error correction coding, with a high E_b/N_0 and a large W, the channel is essentially perfect. For both the coded and uncoded channels, at low spreading factors W, the overall throughput is limited mainly by the channel errors. In this case, the synchronization is nearly ideal, since the duration of a chip T_c is long. Also, for the optimum g^*, the number of radios transmitting is a small fraction of the total M, so the network can be closely approximated by the simpler infinite population model, with a traffic rate $G = Mg$.

The third effect is the synchronization process. A single cycle, single correlator, two stage acquisition circuit is described in Chapter 5. For this circuit, when the number of chips per symbol N is large enough to insure a small probability of bit error, a large fraction of packets are missed due to imperfect synchronization, even for the optimum settings of the synchronization parameters. However, by using a more expensive circuit with many active correlators, or by using a multiple cycle preamble, the performance can be greatly improved. The degradation due to synchronization was as small as 6% for a synchronization circuit of moderate cost and complexity, indicating that this effect can be held to a minimum. Nevertheless, even though the effect of imperfect synchronization can be minimized, the performance

is fairly sensitive to the synchronization parameters, and an improper operating point can cause the performance to degrade significantly.

In addition to the network using a disciplined ALOHA protocol, the performance of a system with ideal channel load sensing was also investigated. The results indicate that at low spreading factors ($W = 64$), the performance can be increased by about 50% by using load sensing. However, an approximate model of propagation delay shows that this improvement degrades to practically nothing as the propagation delay increases towards the transmission time of a packet. Furthermore, the actual performance will be worse because the channel load sensing will not be ideal. Also, in a real PR network, there are likely to be hidden nodes, which will further degrade the performance. Therefore, it seems that the improvement due to channel load sensing is not worth the added cost and complexity of implementing this protocol, especially for systems with higher spreading factors.

The results from the Markovian model demonstrate the impact of the radio channel parameters, the synchronization circuit parameters, and the network parameters on the network throughput and probability of success. Many of the assumptions and approximations that were validated by simulation for the particular system studied will carry over to the study of other channels and synchronization circuits. For example, if the performance of a different synchronization circuit is analyzed in a multi-user environment, the overall probabilities of reception can be determined from the resulting probabilities of detection and of false alarm. These can then be used in the network model to give the throughput. Once the appropriate error rates for the Poisson error model or the probabilities of reception are determined, the Markovian model described in Chapter 3 can be used to find the network throughput for a variety of channels and of synchronization circuits.

7.1 Future Work

As is often the case, although the work completed in this dissertation has answered a number of important questions, it has also brought up many new questions which remain unanswered. Because the model was successful in leading to numerically tractable yet accurate results for the particular system studied, a similar approach should work for a variety of radio channels. For example, the BER of a frequency hopped SSMA channel has been analyzed by a number of authors. Typically, the FH system makes use of m-ary modulation, so that several bits of information are transmitted simultaneously. Therefore, the probabilities of error for neighboring bits is highly correlated. However, it is again the probability of success for the packet, not for individual bits, which is important, and this can be found as the product over all m-ary words of the probability that the entire word was successful. A further difficulty is encountered in slow frequency hopped systems, for which there are many bits per hop. This again leads to correlation from word to word, but the product will still give the lower bound. Consequently, even with m-ary modulation and slow frequency hopping, it is believed that a Poisson error model can be formulated which will give a lower bound on the probability that a packet is successful.

The integrated model might also be extended to other convolutional codes, and also to sequential decoders and block coding. The major hurdle in the study of block codes is the requirement that the error model be Poisson. It may be possible to consider a mini-slotted system, in which the slot size is equal to a single block. For such a system, the interference is constant over the entire block, and the probability of block error is conditionally independent from block to block, given the number of transmissions X. However, this model leads to a discrete time Markov chain, which involves many more transitions than the continuous time chains. Even if the low

probability transitions are removed, it is not clear that the resulting transition rate matrices $\mathbf{Q}$ and $\mathbf{Q}^*_{\mathrm{aux}}$ could be numerically inverted. An alternative is to use the continuous time approximation. For a code rate of roughly 1/2, a block length of 64 symbols corresponds to only 32 bits. The approximation used for the convolutional decoder can then be used, namely, that the probability of block error is only a function of the channel state at the end of the block. Although it is somewhat crude, the approximation can then be made that the errors are generated from a Poisson process, with a rate $\epsilon(X)$ such that

$$e^{-\epsilon(X)T_{block}} = 1 - P_{e,block}, \tag{7.1}$$

where T_{block} is the duration of a block.

The more general case of a non-uniform network requires a substantial reformulation of the network model. If the received power differs according to who is transmitting, the model must account for the state of each individual user, resulting in a greatly expanded state space. This expanded model is also needed for the study of a multi-hop network, in which some PRU's are not in radio connectivity, so must communicate through one or several intermediate radios. From the expanded state, for each state it is known exactly which radios are transmitting to or receiving from which radios. Using this information, tight approximations to the probability of symbol error can be found, taking into account the varying strengths of each interfering signal. It is well recognized that the degradation due to a single strong interfering signal is much worse than that due to many weak interfering signals, even if the total interference power is the same. A model with an expanded state space would indicate the impact of this "near-far effect" on the network throughput.

Unfortunately, the large state space required prohibits the calculation of numerical results for any but the smallest networks. Programming done by the author and

Brazio [7] led to results for general networks with a maximum size of about six radios. A great deal of work was done towards developing an approximate model that could be used for studying these more general networks. However, the approximate models derived were still limited to networks of 10 to 15 radios, but the approximate throughput differed from the exact results for a uniform fully connected network by factors of two or three. Thus, the approximate models of a general network were too inaccurate to be of use. Of course, the results given in this dissertation probably would have been numerically intractable as recently as 15 years ago, so in the near future, the general model with an expanded state space may very well become useful for networks of a more realistic size. Meanwhile, the non-uniform and multi-hop networks might best be studied by considering the behavior of specific restricted cases, such as a network with two classes, each of which has a different transmission power. The Markovian model analyzed in Chapter 4 can be extended to many of these specific systems without growing enormously as it did for a completely general network.

In conclusion, this analysis of the spread spectrum multiple access packet radio network led to numerical results that indicated the specific impact of the channel level parameters on the performance. This also led to the identification of three general effects, and the ranges of the parameters for which each of the effects becomes dominant. Furthermore, the model presented can be extended in a variety of ways. Thus, it is hoped that in the future, a number of other systems will be analyzed based on the foundation of the work presented herein, leading to similar insights regarding the performance of those systems.

Appendix A

Finite Population Throughput

In this appendix, the equation for raw throughput is derived from renewal theory. The state space $Z(t)$ is expanded to explicitly indicate the state of user u. This expanded state space is denoted by $Z_E(t)$.

The following terms are used.

X_E: The number of radios transmitting, not including radio u.

R_E: The number of radios receiving, not including radio u.

$$s_u: \quad \text{the state of user } u, \qquad s_u = \begin{cases} 0, & u \text{ is idle;} \\ 1, & u \text{ is transmitting;} \\ -1, & u \text{ is receiving.} \end{cases}$$

Ω_E: The expanded state space,

$$\Omega_E = \left\{ (X_E, R_E, s_u): \begin{array}{l} 0 \le X_E \le M-1 \\ 0 \le R_E \le X_E + s_u \\ X_E + R_E \le M-1 \\ s_u \in \{-1, 0, 1\}. \end{array} \right\}$$

$\pi_{0E}(X_E, R_E, s_u)$: The steady state probability of being in state (X_E, R_E, s_u).

$S_u(X_E, R_E)$: The fraction of time that user u is successfully transmitting packets which found the network in state $(X_E, R_E, 0)$ upon arrival.

The analysis follows the solution to a more general case for multihop networks found by Brazio and Tobagi in [5]. The times of successive transitions from the state $(X_E, R_E, 0)$ to the state $(X_E, R_E, 1)$ are regeneration points for the Markov chain $Z_E(t)$. Consider the cycles defined by the time intervals between two successive regeneration points. Let $C_k^u(X_E, R_E)$ denote the length of the kth cycle, and let $T_k^u(X_E, R_E)$ denote the successful length of the packet whose arrival initiated cycle k. The sequence

$$\{(C_k^u(X_E, R_E), T_k^u(X_E, R_E)) : k \geq 1\}$$

is a sequence of i.i.d. pairs of random variables. Let $E(C_k^u(X_E, R_E))$ denote the average value of $C_k^u(X_E, R_E)$ and $E(T_k^u(X_E, R_E))$ the average value of $T_k^u(X_E, R_E)$. From the theory of renewal processes,

$$S_u(X_E, R_E) = \frac{E(T_k^u(X_E, R_E))}{E(C_k^u(X_E, R_E))} \qquad \text{with probability 1.} \tag{A.1}$$

Also, as shown by Brazio and Tobagi in [5],

$$E(C_k^u(X_E, R_E)) = \frac{1}{\lambda\, \pi_{0E}(X_E, R_E, 0)}. \tag{A.2}$$

If user u is idle, and there are X_E other radios transmitting and R_E receiving, the total number of users transmitting is X_E, and receiving is R_E. Thus, the event $\{Z_E = (X_E, R_E, 0)\}$ is the same as the event $\{Z = (X, R)$ and u is idle$)\}$, $X = X_E$ and $R = R_E$. In other words,

$$\pi_{0E}(X_E, R_E, 0) = \pi_0(X, R) P_{SI}(X, R) \qquad \begin{matrix} X = X_E \text{ and} \\ R = R_E. \end{matrix} \tag{A.3}$$

so Eqn. A.2 becomes

$$E(C_k^u(X_E, R_E)) = \frac{1}{\lambda \pi_0(X,R) P_{SI}(X,R)} \qquad \begin{matrix} X = X_E \text{ and} \\ R = R_E. \end{matrix} \tag{A.4}$$

Because all users are identical, the expected successful length of a packet that is transmitted given X_E other radios transmitting when it arrived does not depend on which particular radio u the packet arrived at. By definition, the quantity $\overline{T}_s(X,R)$ is conditioned on the packet being received. If the packet is not received, this corresponds to an immediate transition to the state *Failure* in the auxiliary Markov chain. Since $T_k^u(X_E, R_E)$ is not conditioned on the packet being received,

$$E(T_k^u(X_E, R_E)) = \overline{T}_s(X,R)) P_{Rx}(X,R), \qquad \begin{matrix} X = X_E \text{ and} \\ R = R_E. \end{matrix} \tag{A.5}$$

Thus,

$$\begin{aligned} S_u(X_E, R_E) &= \frac{E(T_k^u(X_E, R_E))}{E(C_k^u(X_E, R_E))} \\ &= \lambda \pi_0(X,R) P_{SI}(X,R) P_{Rx}(X,R) \overline{T}_s(X,R). \end{aligned} \tag{A.6}$$

for $X = X_E$ and $R = R_E$, for those states $\{(X_E, R_E) : (X_E, R_E, 0) \in \Omega_E\}$. The sets of packets counted for the cycles defined by different states (X,R) are disjoint, so the user throughput is found by summing the expected contribution to throughput given a starting state (X,R) over all states for which at least one radio is idle. The quantity being summed is non-zero only for states $\{(X,R) : (X+1, R+1) \in \Omega_{\text{aux}}^*\}$, so

$$S_u = \sum_{(X+1,R+1) \in \Omega_{\text{aux}}^*} \lambda \pi_0(X,R) P_{SI}(X,R) P_{Rx}(X,R) \overline{T}_s(X,R). \tag{A.7}$$

Appendix B

Simulation Models

This appendix describes the simulation models used to verify the approximate analyses of the Viterbi decoder and the synchronization process. Some of the modeling assumptions described in Chapter 2 are not necessary in a simulation model (e.g., the exponential distribution for packet lengths). However, the intent of the simulations is to validate specific approximations made, rather than to generate data for a more realistic overall model. Therefore, to the extent possible, most aspects of the process simulated are identical to the analytical treatment.

B.1 FEC Decoder Memoryless Approximation

The first task of the simulation is to verify that the coded channel can be closely approximated by a memoryless process. To this end, the network process simulated is identical to the finite population primary Markov chain of §4.2.1. The one change is that the probability of success for a packet of length τ, starting at

time t_0, is calculated as

$$P_S = \exp\left\{-\int_{t_0}^{t_0+\tau} \epsilon(X'(s))ds\right\}. \qquad (B.1)$$

where

$$\epsilon(X) = -\ln\left(P_C(P_{e,symbol}(X))\right) b, \qquad (B.2)$$

as before, but instead of the current state $X(t)$, $X'(t)$ is used, which is the maximum or minimum number of radios transmitting during the interval $[t - 35T_{bit}, t]$.

A discrete event simulation was used. There are two event lists, one for future events and one for past events. Events are the beginning of transmission, BOT, the end of a transmission which is not received, EOT, and the end of a transmission which is received, EOTR. The lists are ordered according to the times of the events. Also, for the event EOTR, there are variables indicating the length and cumulative probability of success for the packet. The network state is again (X, R).

The mean packet length is set equal to one unit of time, so $1/\mu = 1$ and $\lambda = g$. After each network transition, the next potential start of transmission is drawn from an exponential distribution, at rate $(M - X - R)g$. If this starting time is prior to the next scheduled transition, a packet transmission is inserted as the next event. Otherwise, the starting time is discarded. At the beginning of transmission, it is determined whether or not the packet is received, by drawing a Bernoulli random variable, with a probability $p = 1 - (X + R)/(M - 1)$. The time of the end of transmission is determined by drawing the length from an exponential distribution, and the appropriate event EOT or EOTR is inserted into the event list. If the packet is received, the cumulative probability of success is initialized to 1.

At each event, the probability of success is updated for all packets that are being received. For an event at time t_0, with the previous event having occurred at time

t_{-1}, the quantity

$$\Delta_{t_{-1}}^{t_0} = \int_{t_{-1}}^{t_0} \epsilon(X'(s))ds \qquad (B.3)$$

is found from the current network state and the list of events (transitions) in the past. The cumulative probability of success is multiplied by $\exp\{-\Delta_{t_{-1}}^{t_0}\}$. At the end of a packet, if the packet is received, the throughput is incremented by τP_S, the packet length times the probability of success.

Using ratio estimation techniques, the expected value of

$$\lim_{T\to\infty} \frac{\sum_{j=1}^{J_T} \tau^{(j)} P_S^{(j)}}{T} \qquad (B.4)$$

can be estimated, where the packets $j = 1, 2, \ldots, J_T$ are all those packets which finish in the interval $[0, T]$. The throughput calculated by setting $X'(t)$ equal to the maximum or minimum of $X(t)$ will correspond to the bounds of §5.5. Furthermore, the simulation itself can be checked by setting $X'(t) = X(t)$, and comparing to the analytical results. As seen in Chapter 5, the agreement is well within the sample confidence intervals.

B.2 Preamble Synchronization

A similar approach is used for validating the network approximations with a realistic synchronization process. In this case, in addition to the effect on error rates, the evolution of the interference also affects the probabilities of detection and of false alarm. A number of events that were treated probabilistically in the analysis are specifically accounted for in the simulation. The model makes use of the probabilities of detection and false alarm determined in §6.2.

The detection and verification probabilities and probability of false verification are all calculated from the average of the interference over the correlation time. The model specifically tracks the network evolution over each preamble, eliminating the need for the auxiliary variables. In addition, the model accounts for the state of the active correlator or correlators at each radio, retaining the number that are busy and the times at which they become free. It is again assumed that a false lock will cause the radio to be busy for a fixed length interval of 64 bits.

The one approximation which is retained from the model of Chapter 6 is that the false detections are generated from a Poisson process, whose rate is a function only of the current interference level, not the average interference over a window of length T_{p_1}. A more accurate model which averaged over this window would be even more complex, and would lead to an increase in the run time. Because this window is small in comparison to the time between network transitions, and the probability of false detection changes rather slowly as a function of the interference, this approximation will have negligible impact on the overall performance.

Compared to the simulation described in §B.1, there are many more events happening in a unit time interval for the simulation of the synchronization process. In particular, except when l_1 is large, there are many false detections occurring at every radio. Therefore, for the simulation to be efficient, it necessary to treat the false detections separately from the network transitions. There are again event lists for future and past events, but a new list is added which accounts for false verifications. The state space is greatly expanded, and explicitly indicates the states of individual radios, which are one of the following: R_p, synchronized to a preamble; R, receiving a packet; X_p, transmitting a preamble; X, transmitting a packet, F, false lock, and *Idle*. Also, the events are listed below.

BOP: Beginning of preamble

PD: Preamble detection

LPD: Last preamble detection

EOV: End of verification

EOLV: End of last verification

EOP: End of preamble

EOT: End of transmission

EOTR: End of received transmission

EOFV: End of false verification

EOFL: End of false lock

Associated with each event are a number of variables. These include the source and destination and the time of the event. For each ongoing preamble, a list is kept of all radios which are receiving the preamble, needed for the case of space homogeneous preamble codes. Also, the cumulative interference is stored for every preamble detection and end of verification, and every end of false verification. For the end of received transmissions, the cumulative probability of success is stored as before. Other variables track the state of each radio, the total number of radios in a given state, and the state of the bank of active correlators at each receiver.

Similar to the updating of the cumulative probability of success described in §B.1, at each event the cumulative noise is updated for all preamble detections, end of verifications, and end of false verifications. New transmissions are drawn from an exponential distribution, at rate $(M - X - R - X_p - R_p - F)g$. If the transmission precedes the next event, it is added to the future event list as the next event; otherwise it is discarded. However, because the false events are treated separately, this transmission is tentative, and may be discarded if the source is in false lock at

the scheduled time. This model accounts for the particular radios involved, so when a new transmission occurs, the source is determined from a random variable which is drawn uniformly over all the idle radios, and the destination is chosen uniformly over all other radios.

After checking for new packets, the false detections occurring before the next event are determined. For each PRU that is not transmitting, the number of false detections is drawn from a Poisson process, at a rate equal to

$$\lambda_{fa_1} = \frac{P_{fa_1}(I)}{T_c}(t_1 - t_0), \tag{B.5}$$

where $I = X + X_p - 1$, and t_0 is the time of the current event and t_1, the time of the next event. The times of the false detections are then chosen uniformly in the interval $[t_0, t_1]$. An ordered list keeps track of the state of the bank of active correlators. For each false detection, it is determined whether there was an active correlator available. If so, an active correlator is assigned to that false detection, and the time of the end of false verification is calculated. The list of active correlators for that radio is updated and an end of false verification event is added to the false event list. This is repeated for every radio that is not transmitting.

Next, the end of false verifications are processed for the interval $[t_0, t_1]$. If the false verification began before time t_0, the interference is updated from the time of the current event, t_0, to the time of the false verification. Otherwise, the interference is constant and equal to $I = X + X_p - 1$. In either case, the probability of false verification is determined using the average interference over k_2 chips. If a false verification occurs, all of the false detections for that radio at times before the end of false lock are deleted. A false lock can have two effects: it can cause a radio to skip a transmission at a scheduling point or it can cause a radio to miss a reception. Because the next event precedes or equals the next scheduling point for all idle

radios, any false lock that ends before the next event does not affect the network behavior. If the false lock continues beyond the next event at time t_1, the end of false lock event is added to the future event list, and the state of the radio is changed to FL. The next event is then checked, to see if it is a new transmission by the radio in false lock. If so, the beginning of preamble event is changed to a null event, which is later removed. This null event is needed so that the next interval over which the false events are calculated is correct.

For the beginning of a preamble, at time t_0, the state of the source is updated to X_p, and the active correlators are all set to idle. Preamble detection events are added to the future event list at the times at which detections may occur, which is simply $t_0 + T_{p_1}$ for the single cycle case. The cumulative interference is initialized to zero. Also, the event of the end of preamble is added to the event list. The final cycle is treated slightly differently, so the event is denoted as the last preamble detection LPD.

At each PD or LPD event, the average interference is found from the cumulative interference divided by T_{p_1}. The probability of detection is determined, and a Bernoulli r.v., with $p = P_{d_1}$ is drawn. If detection is successful, and the state of the destination is *Idle*, and there is an active correlator available at the destination, the active correlator list is updated and the list of radios verifying the preamble is initialized to include the destination. For space homogeneous preambles, this is repeated for all idle radios. If the list of radios verifying is not empty, an end of verification event is added to the future event list. The cumulative interference for the verification is initialized to 0. If the detection event was a LHD, rather than adding an EOV event, the EOP event is changed to an EOLV.

When the verification finishes, at an EOV or EOLV event, the average interference is again found, and it is determined probabilistically wheter the preamble is

verified or not, for all radios verifying. If a radio successfully verifies the preamble, the state of that radio is changed to R_p, and it is recorded which source is being received. For receiver directed codes, only the destination can be verifying, so only the destination can lock onto the preamble.

At the end of the preamble, the length τ of the packet is drawn from an exponential distribution. If the length is less than T_p then the data portion is set to length zero. Otherwise, the length of the data portion is set equal to $\tau - T_p$. If the state of the destination is R_p, and it is receiving the preamble from the correct source, an EOTR event is added to the event list. Otherwise, an EOT event is added. The cumulative probability of success is set to one. Next, all those radios that were locked onto the preamble but are not the destination, in the space homogeneous case, are returned to the state *Idle*. The end of transmission and end of received transmission are similar to the events discussed in §B.1.

This simulation has been used to validate the approximate model of the SSMA PR network with a realistic preamble synchronization process. The model is complex, but is able to specifically account for all the events affecting the synchronization. As seen in Chapter 6, the results from simulation agree well with the approximate analytical results.

Bibliography

[1] N. Abramson, "The ALOHA System — Another Alternative for Computer Communications," *AFIPS Conference Proc.*, 1970 Fall Joint Computer Conference, Vol. 37, pp. 281-285, 1970.

[2] S. Bellini and F. Borgonovo, "On the Throughput of an ALOHA Channel with Variable Length Packets," *IEEE Trans. Comm.*, Vol. COM-28, No. 11, pp. 1932-1935, November 1980.

[3] R. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Ma., 1983.

[4] R. Boorstyn and A. Kershenbaum, "Throughput Analysis of Multihop Packet Radio," *Proc. ICC '80*, Seattle, Washington, June 1980.

[5] J. Brazio and F. Tobagi, "Theoretical Results in Throughput Analysis of Multihop Packet Radio Networks," *Proc. ICC '84*, Amsterdam, May 1984.

[6] J. Brazio and F. Tobagi, "Throughput Analysis of Spread Spectrum Multihop Packet Radio Networks," *Proc INFOCOM '85*, Washington, D.C., March, 1985.

[7] J. Brazio, "Capacity Analysis of Multihop Packet Radio Networks under a General Class of Channel Access Protocols and Capture Modes," Ph.D. dissertation, Dept. of Elect. Eng., Stanford Univ., Stanford, Calif., 1986.

[8] R. Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, N.J., 1973.

[9] M. Chen and R. Boorstyn, "Throughput Analysis of Code Division Multiple Access (CDMA) Multihop Packet Radio Networks in the Presence of Noise," *Proc INFOCOM '85*, Washington, D.C., March, 1985.

[10] G. Clark, Jr. and J. Cain, *Error Correction Coding for Digital Communications*, Plenum Press, New York, 1981.

[11] D. Davis and S. Gronemeyer, "Performance of Slotted ALOHA Random Access with Delay Capture and Randomized Time of Arrival," *IEEE Trans. Comm.* Vol. COM-28, No. 5, pp. 703-710, May 1980.

[12] O. deSouza, M. Chen, R. Boorstyn, "A Comparison of the Performance of Protocols in Packet Radio Networks," *Proc. MILCOM '85*, Boston, October 1985.

[13] J. DiFranco and W. Rubin, *Radar Detection*, Prentice-Hall, Englewood Cliffs, N.J., 1968.

[14] R. Dou and L. Milstein, "Error Probability Bounds and Approximations for DS Spread Spectrum Communication Systems with Multiple Tone or Multiple Access Interference," *IEEE Trans. Comm.* Vol. COM-32, No. 5, pp. 493-502, May 1984.

[15] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 1, Wiley, New York, 1968.

[16] M. Ferguson, " A Study of Unslotted ALOHA with Arbitrary Message Lengths," *DATA COMM.*, P.Q., Canada, October 1975.

[17] J. Fischer, et. al., "Wideband Packet Radio Technology," *IEEE Journ. Selected Areas in Comm.*, to be published, 1986.

[18] M. Flynn, C. Spangler, and A. Zimmerman, "The Stanford Packet Radio Network," *Proc. COMPCON '86*, San Francisco, 1986.

[19] E. Geraniotis and M. Pursley, "Error Probability for Direct Sequence Spread Spectrum Multiple Access Communications — Part II: Approximations," *IEEE Trans. Comm.* Vol. COM-30, No. 5, pp. 995-995, May 1982.

[20] E. Geraniotis, "Coherent Hybrid DS-SFH Spread Spectrum Multiple Access Communications," *IEEE Journ. Selected Areas in Comm.*, Vol. SAC-3, No. 5, pp. 695-705, September 1985.

[21] E. Geraniotis, "Performance of Noncoherent Direct Sequence Spread Spectrum Multiple Access Communications," *IEEE Journ. Selected Areas in Comm.* Vol. SAC-3, No. 5, pp. 687-694, September 1985.

[22] J. Heller and I. Jacobs, "Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. Comm.* Vol. COM-19, No. 5, pp. 835-848, October 1971.

[23] R. Kahn, S. Gronemeyer, J. Burchfiel, R. Kunzelman, "Advances in Packet Radio Technology," *Proc. of the IEEE* Vol. 66, No. 11, pp. 1468-1496. November, 1978.

[24] D. Kincaid, et. al, "ITPACK 2C: A FORTRAN Package for Solving Large Sparse Linear Systems by Adaptive Accelerated Iterative Methods," *ACM Trans. on Math. Softw.*, Vol. 8, No. 3, pp. 302-322, September, 1982.

[25] L. Kleinrock, *Queueing Systems*, Vol. 1, Wiley, New York, 1975.

[26] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I — Carrier sense multiple access modes and their throughput delay characteristics," *IEEE Trans. Comm.*, Vol. COM-23, No. 12, pp. 1400-1416, December 1975.

[27] K. Klemba, D. Nielson, J. Tornow, and B. Leiner , "Packet Radio Network Executive Summary," Defense Advanced Research Projects Agency, IPTO-83-7, July, 1983.

[28] M. Kowatsch, " Synchronization in a Spread Spectrum Communication Modem Based on SAW Convolvers," *Proc. MILCOM '84*, Los Angeles, October, 1984.

[29] D. Laforgia, A. Luvison, V. Zingarelli, "Bit Error Rate Evaluation for Spread Spectrum Multiple Access Systems," *IEEE Trans. Comm.* Vol. COM-32, No. 6, pp. 660-669, June 1984.

[30] Linkabit Corp, Final Rep., "Coding systems study for high data rate telemetry links," Contract NAS2-6024, NASA Ames Res. Ctr. Rep. CR-114278, Moffett Field, Calif., 1971.

[31] J. Musser and J. Daigle, "Throughput Analysis of an Asynchronous Code Division Multiple Access (CDMA) System," *Proc ICC '82*, Philadelphia, June, 1982.

[32] R. Nelson and L. Kleinrock, "The Spatial Capacity of a Slotted ALOHA Multihop Packet Radio Network with Capture," *IEEE Trans. Comm.* Vol. COM-32, No. 6, pp. 684-694, June 1984.

[33] Albert. H. Nutall, "Numerical Evaluation of Cumulative Probability Distribution Functions Directly from Characteristic Functions," *Proc. IEEE*, Nov. 1969, pp. 2071-2072.

[34] J. Odenwalder, "Optimum decoding of convolutional codes," Ph.D. dissertation, Syst. Sci. Dept., Univ. California, Los Angeles, 1970.

[35] A. Polydorous and C. Weber, "A Unified Approach to Serial Search Spread Spectrum Code Acquisition — Part I: General Theory," *IEEE Trans. Comm.* Vol. COM-32, No. 5, pp. 542-549, May 1984.

[36] A. Polydorous and C. Weber, "A Unified Approach to Serial Search Spread Spectrum Code Acquisition — Part II: A Matched Filter Receiver," *IEEE Trans. Comm.* Vol. COM-32, No. 5, pp. 550-560, May 1984.

[37] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, *Numerical Recipes*, Cambridge University Press, New York, N.Y., 1986.

[38] M. Pursley, "Performance Evaluation for Phase Coded Spread Spectrum Multiple Access Communications — Part I: System Analysis," *IEEE Trans. Comm.* Vol. COM-25, No. 8, pp. 795-799, August 1977.

[39] M. Pursley, D. Sarwate, W. Stark, "Error Probability for Direct Sequence Spread Spectrum Multiple Access Communications — Part I: Upper and Lower Bounds," *IEEE Trans. Comm.* Vol. COM-30, No. 5, pp. 975-984, May 1982.

[40] M. Pursley, "Throughput of Frequency Hopped Spread Spectrum Communications for Packet Radio Networks," *Proc. 1983 CISS*, John Hopkins Univ., March 1983.

[41] M. Pursley, "Frequency-hop Transmission for Satellite Packet Switching and Terrestrial Packet Radio Networks," *IEEE Trans. Info. Theory*, 1986 (to appear); also, *Coordinated Science Laboratory Report*, Report T-144, University of Illinois, June 1984.

[42] M. Pursley, "Spread Spectrum Multiple Access Communications," pp. 139-199 in *Multi-User Communication Systems*, G. Longo, Ed. Springer-Verlag, Vienna and New York, 1981.

[43] D. Raychaudhuri, "Performance Analysis of Random Access Packet Switched Code Division Multiple Access Schemes," *IEEE Trans. Comm.* Vol. COM-29, No. 6, pp. 895-900, June 1981.

[44] S. Rappaport and D. Schilling, "A Two Level Coarse Code Acquisition Scheme for Spread Spectrum Radio," *IEEE Trans. Comm.* Vol. COM-28, No. 9, pp. 895-900, September, 1980.

[45] R. Scholtz, "The Origins of Spread Spectrum Communications," *IEEE Trans. Comm.* Vol. COM-30, No. 8, pp. 822-854, May 1982.

[46] R. Scholtz, "The Spread Spectrum Concept," *IEEE Trans. Comm.* Vol. COM-25, No. 8, pp. 748-755, August 1981.

[47] E. Sousa and J. Silvester, "Spreading Code Protocols for Distributed Spread Spectrum Packet Radio Networks," *IEEE Trans. Comm.*, 1987 (to appear).

[48] E. Siess and C. Weber, " Acquisition of Direct Sequence Signals with Modulation and Jamming," *IEEE Journ. Selected Areas in Comm.*, Vol. SAC-4, No. 2, pp. 254-271, March 1986.

[49] M. Simon, J. Omura, R. Scholtz, B. Levitt, *Spread Spectrum Communications*, Vol. 1, 2, and 3, Computer Science Press, Rockville, Md., 1985.

[50] D. Taipale, "Direct Sequence Spread Spectrum Packet Radio Performance," *Coordinated Science Laboratory Report*, Report T-155, University of Illinois, November 1984; also, M. Pursley and D. Taipale, "Performance of Spread Spectrum Packet Radio with Convolutional Codes and Viterbi Decoding," *IEEE Trans. Comm.*, submitted for publication, Dec. 1985.

[51] A. Viterbi, "Convolutional Codes and their Performance in Communication Systems," *IEEE Trans. Comm.* Vol. COM-19, No. 5, pp. 751-772, October 1971.

[52] C. Weber, G. Huth, and B. Batson, "Performance Considerations of Code Division Multiple Access Systems," *IEEE Trans. Veh. Tech.* Vol. VT-30, No. 1, pp. 3-10, January 1981

[53] K.Yao, "Error Probability of Asynchronous Spread Spectrum Multiple Access Communication Systems," *IEEE Trans. Comm.* Vol. COM-25, No. 8, pp. 803-809, August 1977.